



Titre: Intégration de modèles de caractéristiques (features) à une
modélisation géométrique pour supporter l'analyse métiers

Auteur: Aurélien Riou

Date: 2007

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Riou, A. (2007). Intégration de modèles de caractéristiques (features) à une
modélisation géométrique pour supporter l'analyse métiers [Master's thesis,
École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/8107/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8107/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

INTÉGRATION DE MODÈLES DE CARACTÉRISTIQUES (FEATURES) À UNE
MODÉLISATION GÉOMÉTRIQUE POUR SUPPORTER L'ANALYSE MÉTIERS

AURÉLIEN RIOU

DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
DÉCEMBRE 2007



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-36935-7

Our file Notre référence

ISBN: 978-0-494-36935-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

INTÉGRATION DE MODÈLES DE CARACTÉRISTIQUES (FEATURES) À UNE
MODÉLISATION GÉOMÉTRIQUE POUR SUPPORTER L'ANALYSE MÉTIERS

présenté par: RIOU Aurélien

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. BALAZINSKI Marek, Ph.D., président

M. MASCLE Christian, Doctorat ès sciences, membre et directeur de recherche

M. PELLERIN Robert, Ph.D., membre

REMERCIEMENTS

Je tiens à remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de cette maîtrise.

Je souhaite exprimer toute ma reconnaissance à mon directeur de recherche, Monsieur Christian Mascle, pour son aide, ses conseils et son support financier durant cette maîtrise.

Je remercie également les membres du jury, Messieurs Marek Balazinski et Robert Pellerin, qui m'ont fait l'honneur d'être les rapporteurs de cette recherche.

Je souhaite aussi remercier Monsieur Valentin Dehan qui m'a aidé, lors de son stage au laboratoire de génie mécanique de l'école, à avancer dans le développement de notre logiciel.

Je voudrais également remercier l'ensemble des étudiants du laboratoire : Yacine Boucekine, Assen Petrov Tchernev, Chunfang Zhang, Ahmad Haidar, Julien Gosse, Abdelhak Nafi pour leur contribution à l'ambiance dans le labo et pour la découverte de toutes ces spécialités culinaires.

Enfin, je remercie ma famille et mes amis de m'avoir soutenu tout au long de cette Maîtrise ès science appliquées.

RÉSUMÉ

Les technologies informatiques ont pris au cours des dernières années une place très importante dans le processus de conception de produit, dans le but d'accroître la compétitivité (réduire les coûts de fabrication et les délais de mise en marché, répondre à la mondialisation, améliorer la qualité des produits, etc).

Ce mémoire de maîtrise s'intéresse à l'importance de l'intégration de l'information au sein du processus de développement de produit pour en supporter l'analyse « métier ». Dans le contexte d'ingénierie simultanée, une grande partie de l'information technologique est perdue à cause d'un manque d'intégration à la structure topologique du produit. Cela entraîne des pertes de productivité. La problématique principale est axée sur l'amélioration de la communication entre les différents acteurs du cycle de vie afin de contribuer à améliorer le cycle de développement du produit.

Cette étude a permis développer une structure de données, basée sur la topologie du modèle B-rep, capable de stocker cette information « métier » relative à chaque entité géométrique et de la véhiculer entre les différents intervenants tout au long du cycle de vie du produit. Cette structure de données spécifique nous permet par la suite de traiter l'information technologique et d'en déduire les améliorations à apporter à la conception du produit pour améliorer les paramètres de son propre cycle de vie. Nous avons décidé de limiter cette étude du processus d'élaboration du brut jusqu'au processus d'assemblage.

Les travaux réalisés dans le cadre de ce mémoire sont abordés dans deux articles soumis à des revues scientifiques :

- “Data structure applied to product lifecycle”, soumis à *Research in Engineering Design – Springer*. Cet article expose la structure de données permettant de

répondre à la problématique de l'étude en rendant l'information numériquement accessible tout au long du cycle de développement.

- “Assisting designer using feature modeling for lifecycle”, soumis à *Computer-Aided Design – Elsevier*. Cet article présente, dans la continuité de l'article précédent, l'utilisation de l'information technologique relative aux entités du produit afin de déterminer la validité des choix du concepteur et de gérer les incompatibilités entre certains paramètres des processus de production.

Une dernière étape discute de l'intégration de la solution logicielle développée dans un contexte industriel. Nous abordons également les contraintes liées au transfert de l'information technologique par rapport à l'évolution du modèle, dans un contexte d'ingénierie simultanée, et les problèmes de l'utilisation de cette méthode sur la tâche des intervenants.

ABSTRACT

During the past few years, the use of computer-based technologies is increasing for the design process, with the intention of augment competitiveness (reduce manufacturing costs and time to market, consider globalisation, improve product quality, etc).

This development work concentrates on the information integration importance throughout the product development process to consider advanced analyse. Regardless of the concurrent engineering concept, the main part of technological information is lost because of a lack of integration to the product topological structure. It leads productivity loss. The main problematic underlines technological information integration of the product, in order to improve communication between the various actors of the product lifecycle.

This study has allowed to develop a data structure, based on the B-rep model topology, able to stock this specific information of geometric entities. This structure eases thus the accessibility of information at each stakeholder throughout the product lifecycle. It also allows to process technological data and highlights non-optimised parameters, in order to improve the product development process. We have decided to limit this study from the blank development process to the assembly process.

Development works are got onto two papers submitted to specialised journals:

- “Data structure applied to product lifecycle”, submitted to *Research in Engineering Design – Springer*. This paper deals with the data structure allowing to ease accessibility of information throughout the development cycle
- “Assisting designer using feature modeling for lifecycle”, submitted to *Computer-Aided Design – Elsevier*. This paper deals with the use of technological information concerning geometric product entities, in order to

verify the viability of designing choices and to manage incompatibilities between some production processes characteristics. This paper is based on the data structure build in the first one.

The last section discuss about integration of this method in industrial solutions. It also deals with the study of technological information transfer considering product changes in a concurrent engineering concept, and the problems on the user's work generated by this method.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX.....	xiii
LISTE DES FIGURES.....	xiv
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
LISTE DES ANNEXES.....	xix
INTRODUCTION	1
 CHAPITRE 1 : REVUE DE LITTÉRATURE	 4
1.1 Ingénierie simultanée	4
1.1.1 Définition et objectifs.....	4
1.1.2 Développement intégré	5
1.2 Conception de produit.....	6
1.2.1 Modélisation du produit	6
1.2.2 Vues en conception	7
1.2.3 Conception pour la fabrication.....	9
1.2.4 Conception pour l'assemblage	10
1.2.5 Conception pour X	11
1.3 Approche par feature.....	11
1.3.1 Définitions.....	12
1.3.2 Représentation mathématique	14
1.3.3 Propriétés des features	14
1.3.4 Utilité des features.....	15
1.3.5 Utilisation des features.....	15

1.3.6 Les types de features	18
1.3.7 Taxinomie des features	24
1.4 Conclusion	26
CHAPITRE 2 : SYNTHÈSE.....	28
2.1 Problématiques.....	28
2.1.1 Intégration de l'information	28
2.1.2 Aide à la conception.....	30
2.1.3 Limites	31
2.2 Démarche et objectifs de recherche	31
2.3 Définition des modèles géométriques	33
2.3.1 Le modèle CSG.....	33
2.3.2 Le modèle B-rep.....	33
2.3.3 Choix du modèle	34
2.4 L'organisation des données topologiques	36
2.5 Le formalisme NIAM.....	37
2.6 Conclusion	39
CHAPITRE 3 : DATA STRUCTURE APPLIED TO PRODUCT LIFECYCLE.....	40
Abstract:	40
Keywords:	41
3.1 Introduction.....	41
3.1.1 Related works.....	42
3.1.2 NIAM Formalism.....	45
3.2 Modeling of the product lifecycle.....	47
3.2.1 The manufacturing process	47
3.2.2 The assembly process.....	50
3.3 The organization of the topological data.....	52
3.3.1 Topology of the part.....	52
3.3.2 Construction of the topological structure.....	53

3.4 Software modeling	60
3.4.1 Modeling of a component in the data structure.....	60
3.4.2 Software purpose.....	61
3.4.3 Prospects	63
3.5 Conclusion	63
3.6 References	64
CHAPITRE 4 : REPRÉSENTATION DES DONNÉES « MÉTIER »	67
4.1 Procédés de développement de produit.....	67
4.2 Implémentation logicielle des features technologiques	69
4.3 Conclusion	73
CHAPITRE 5 : ASSITING DESIGNER USING FEATURE MODELING FOR LIFECYCLE	74
Abstract:	74
Keywords:	75
5.1 Introduction	75
5.2 Related works.....	77
5.2.1 Feature modelling.....	78
5.2.2 Aided decision.....	79
5.3 Information flows.....	81
5.4 Data structure	84
5.4.1 Part and product data structure.....	85
5.4.2 Lifecycle process data structure	89
5.5 Integrated Engineering.....	92
5.5.1 Decision aid.....	92
5.5.2 Incompatibility management.....	97
5.6 When add data?	100
5.7 Conclusion	101
5.8 References	102

CHAPITRE 6 : DISCUSSION GÉNÉRALE	106
6.1 Information « métier » liée au feature durant le cycle de développement	106
6.2 Intégration dans un contexte industriel	107
6.3 Échange de l'information	108
6.4 Discussion sur le suivi de l'information selon l'évolution géométrique du produit	111
6.5 Gêne de l'utilisateur	114
6.6 Conclusion	115
CONCLUSION	116
BIBLIOGRAPHIE	119
ANNEXES	123

LISTE DES TABLEAUX

Tableau 5.1 Compatibility between processes and materials, adapted from [1].....	99
--	----

LISTE DES FIGURES

Figure 1.1 Impact du changement au cours du cycle de développement.....	5
Figure 1.2 Principe de l'ingénierie simultanée, adapté de (Sky, 1996).....	6
Figure 1.3 Les différentes vues d'un produit, adapté de (Tollenaere, 1998).....	8
Figure 1.4 Interprétation d'un <i>feature</i> suivant le point de vue de conception (a) ou de fabrication (b), adapté de (Deneux, 2002)	9
Figure 1.5 Modèle de <i>feature</i> associé aux caractéristiques géométriques, adapté de (Chang, Wang, & Wysk, 2006).....	16
Figure 1.6 Exemple d'un arbre de spécification d'un produit modélisé par <i>feature</i>	17
Figure 1.7 Mode d'utilisation des <i>features</i> , d'après (Shah, 1991) et (Deneux, 2002).....	18
Figure 1.8 Exemples de <i>features</i> relatifs à une bielle, adapté de (Tollenaere, 1998)	19
Figure 1.9 Taxinomie des <i>features</i> de tolérance, adapté de (Tollenaere, 1998)	25
Figure 1.10 Taxinomie des <i>features</i> d'usinage en tournage, adapté de (Duverlie, 1996).....	26
Figure 2.1 Arbre de modélisation CSG d'un solide (à droite)	33
Figure 2.2 Intégration des données technologiques à la structure géométrique	35
Figure 2.3 Topologie de la pièce (modélisation B-rep)	37
Figure 2.4 Représentation graphique du formalisme NIAM	38
Figure 2.5 Exemples du formalisme NIAM.....	39
Figure 3.1 Schematical representation of NIAM formalism.....	46
Figure 3.2 Examples of NIAM formalism	47
Figure 3.3 Schematical representation of the manufacturing process.....	48
Figure 3.4 Partial expanding of a process sub-object	49
Figure 3.5 Schematical representation of the assembly process	51
Figure 3.6 Schematical representation of the product topological structure.....	54
Figure 3.7 Schematical representation of the material feature.....	55
Figure 3.8 Schematical representation of the geometrical tolerance feature	57
Figure 3.9 Schematical representation of the dimensional tolerance feature.....	58

Figure 3.10 Partial schematical representation of the surface treatment feature	59
Figure 3.11 Schematical representation of the surface quality feature	59
Figure 3.12 Modeling of the data structure in the software	60
Figure 3.13 Features view used for data entry	61
Figure 3.14 Surface roughness features view	62
Figure 3.15 Tolerance features view	63
Figure 4.1 Taxinomie du procédé d'usinage	68
Figure 4.2 Exemple de structure générée grâce à la librairie OCAF	70
Figure 4.3 Représentation du <i>feature</i> d'état de surface dans la base de données	71
Figure 5.1 Integration of total engineering knowledge	77
Figure 5.2 Data integration in the development process	82
Figure 5.3 Feature views of the product	83
Figure 5.4 Information exchange in conventional (a) and integrated (b) design	84
Figure 5.5 Schematized product topology	86
Figure 5.6 Intrinsic and extrinsic characteristics of the face	87
Figure 5.7 Tree structure of a part (a) and an assembly (b)	89
Figure 5.8 Modeling of lifecycle processes	90
Figure 5.9 Example of subdivision of processes	91
Figure 5.10 Subdivision of positioning attribute according to the process of machining (a) or assembly (b)	92
Figure 5.11 Decision aided principle	93
Figure 5.12 Efficiency assessment of the product	94
Figure 5.13 General cost relationship of various degrees of accuracy for a number of machining processes, adapted from [30]	95
Figure 5.14 Schematically representation of the efficiency factor	96
Figure 5.15 Example of user interface for aided decision	97
Figure 5.16 User interface using strict management of incompatibility	98
Figure 5.17 User interface using large management of incompatibility	100
Figure 5.18 Graphical representation of time-saving	101

Figure 6.1 Extrait d'un fichier de données utilisant le format STEP	110
Figure 6.2 Exemple de <i>feature</i> technologique au format STEP	111
Figure 6.3 Suivi de l'information avec l'évolution géométrique du produit	112

LISTE DES SIGLES ET ABRÉVIATIONS

B-Rep	: Boundary Representation
CAD	: Computer Aided Design
CAM	: Computer Aided Manufacturing
CAO	: Conception Assistée par Ordinateur
CE	: Concurrent Engineering
CFAO	: Conception et Fabrication Assistée par Ordinateur
CIM	: Computer-Integrated Manufacturing
CSG	: Constructive Solid Geometry
DFA	: Design For Assembly
DFM	: Design For Manufacturing
DFMA	: Design For Assembly and Manufacturing
DFX	: Design For X
DMU	: Digital Mock-Up
FBM	: Feature Based Modeling
IPD	: Integrated Product Development
IT	: Intervalle de Tolérance

LMC	: Least Material Condition
MMC	: Maximum Material Condition
NIAM	: Nijssen Information Analysis Method
OCAF	: Open Cascade Application Framework
PLM	: Product Lifecycle Management
RFS	: Regardless of Feature Size
STEP	: Standard for Exchange of Product data

LISTE DES ANNEXES

Annexe A : Tolérancement de l'entité Face.....	123
Annexe B : Définition technologique de l'entité face.....	124
Annexe C : Caractéristiques du feature de traitement de surface	125

INTRODUCTION

Dans un contexte de compétitivité mondiale, les entreprises se doivent d'améliorer leur productivité en diminuant les coûts et les délais de mise sur le marché, sans pour autant réduire la qualité des produits. Les logiciels de développement de produits sont en constante évolution depuis leur apparition, et contribuent grandement à ces objectifs.

Les compagnies ont de plus en plus besoin de faciliter le transfert d'information entre les intervenants du processus de développement de produit afin d'améliorer la production. Les concepts d'ingénierie simultanée et de développement de produit intégré ont permis de prendre en considération plusieurs étapes du cycle de conception. Mais cette approche permettant un développement et des prises de décisions en parallèle dépend fortement de la gestion d'une information interdisciplinaire. À terme, l'idéal de ce concept serait d'intégrer l'ensemble du cycle de vie du produit, de l'étude préliminaire au recyclage en fin de vie, en passant par la conception, la fabrication, l'assemblage et la maintenance.

Les outils de gestion du cycle de vie du produit ("*Product Lifecycle Management*" – PLM) permettent actuellement de gérer la transmission d'information relative à la pièce et à l'assemblage entre tous les acteurs du processus.

En étudiant le cycle de développement du produit, on constate cependant qu'une grande quantité d'information est exploitée lors de la conception, mais qu'une petite partie seulement est consignée dans la maquette numérique. En effet, lors de l'élaboration du produit, le concepteur effectue un grand nombre d'études relatives au cycle de vie (fabricabilité, assemblabilité, maintenabilité, recyclabilité, facteur humain, etc) qui l'amènent à faire des choix de conception. La géométrie du composant caractérise l'ensemble de ces décisions, qui resteront implicites tout au long du cycle de vie du

produit. Pourtant, ces informations sont nécessaires dans les étapes suivant la conception pour déterminer certaines caractéristiques des procédés d'élaboration.

De la même façon, on estime que le travail de l'ingénieur de conception représente une très faible part du travail global pour l'élaboration du produit, alors que les décisions qu'il prend entraînent plus de 70% des coûts de fabrication. Le concept de conception pour l'industrialisation prend alors tout son sens, pour aider le concepteur dans sa démarche.

Les premières méthodologies, introduites par Boothroyd-Dewhurst et Hitachi, concernaient la conception pour l'assemblage, afin d'optimiser le processus de production. Ces méthodes ont ensuite été généralisées au domaine de la fabrication.

Dans notre approche, les données techniques pertinentes sont attachées à la structure de données topologiques du produit. Des algorithmes extraient automatiquement l'information pour définir l'efficience des différents procédés d'élaboration. Le concepteur étant confronté à plusieurs solutions technologiques, il peut déterminer le design optimal en prenant en compte ses impacts sur le reste du cycle de développement.

Les travaux présentés dans ce mémoire sont axés autour de deux articles scientifiques "Data structure applied to product lifecycle" et "Assisting designer using feature modeling for lifecycle", présentés aux chapitres 3 et 5. Ces articles ont servi de support à l'implémentation logicielle à l'aide de la plateforme CAO de développement OpenCASCADE.

Ce mémoire s'articule autour de 6 chapitres :

- Le chapitre 1 est une revue de littérature positionnant la problématique du mémoire parmi les travaux de recherche actuels en conception de produits et en utilisation des *features* technologiques.

- Le chapitre 2 est une synthèse présentant la démarche de l'ensemble du travail de recherche et l'organisation générale du mémoire. Il spécifie les problématiques de l'étude et la cohérence des articles par rapport aux objectifs de la recherche.
- Le chapitre 3 est un article qui présente la structure de données utilisée tout au long de cette étude pour faciliter le transfert d'information au sein du processus d'élaboration du produit. L'implémentation logicielle est également partiellement détaillée.
- Le chapitre 4 sert de transition entre les deux articles. Il complète la présentation de la structure de données et l'implémentation logicielle qui seront à la base de l'extraction des données dans le chapitre 5.
- Le chapitre 5 est un article qui présente l'approche d'aide à la conception développée au cours de cette étude. Il explique la méthodologie d'extraction des informations technologiques de la structure de données du produit et leur traitement en vue de renseigner le concepteur sur la viabilité de ses choix technologiques.
- Le chapitre 6 discute de certains aspects méthodologiques et résultats en lien avec la revue de littérature. Il traite notamment de l'intégration de la méthodologie dans un contexte industriel et des problèmes liés à l'évolution géométrique du produit au niveau du transfert des données techniques. Il analyse enfin les contraintes occasionnées par l'application de cette méthode.

La conclusion expose les contributions du mémoire à l'avancement des connaissances et au développement des technologies. Elle introduit également les perspectives et nouvelles voies de recherche qui découlent de cette étude.

CHAPITRE 1 : REVUE DE LITTÉRATURE

L'objectif de cette revue de littérature est de positionner la problématique du mémoire parmi les travaux de recherche actuels. Ce mémoire est présenté par articles, une revue de littérature spécifique aux domaines abordés dans chaque article est donc incluse aux chapitres 3 et 5. Ce chapitre présente les concepts d'ingénierie simultanée, de conception de produit et, enfin, aborde la notion de feature (ou caractéristique métier).

1.1 Ingénierie simultanée

1.1.1 Définition et objectifs

L'ingénierie simultanée (concourante ou collaborative, "*Concurrent Engineering*" – CE) est une approche systématique qui prend en considération simultanément les phases du cycle de vie du produit : identification des besoins du client, conception du produit, processus de fabrication, fonctionnement, maintenance, recyclage. Cette approche basée sur un partage des données multidisciplinaires permet aux différents intervenants (concepteur, ingénieur méthode, responsable marketing, etc.) de travailler en parallèle, donc de diminuer les délais de conception et de fabrication tout en facilitant la réactivité et les changements, etc.

Si cette approche permet de réduire les erreurs et les itérations ainsi que d'augmenter la qualité du produit, elle est en revanche très contraignante sur le plan organisationnel : suivi des tâches très important, difficultés de compatibilité entre les logiciels, investissements importants, sécurité des informations limitée, etc.

1.1.2 Développement intégré

L'ingénierie simultanée nécessite l'introduction du développement de produit intégré ("*Integrated Product Development*" – IPD), répondant aux diverses contraintes. Ce processus inclut tous les aspects du cycle de vie de la conception à la livraison du produit (qualité, coûts, délais, exigences du client, etc.). Il permet une définition progressive (Figure 1.1) des éléments et notamment le raffinement de différentes spécifications en analysant les influences sur l'ensemble du produit (réduction du poids, augmentation de la performance).

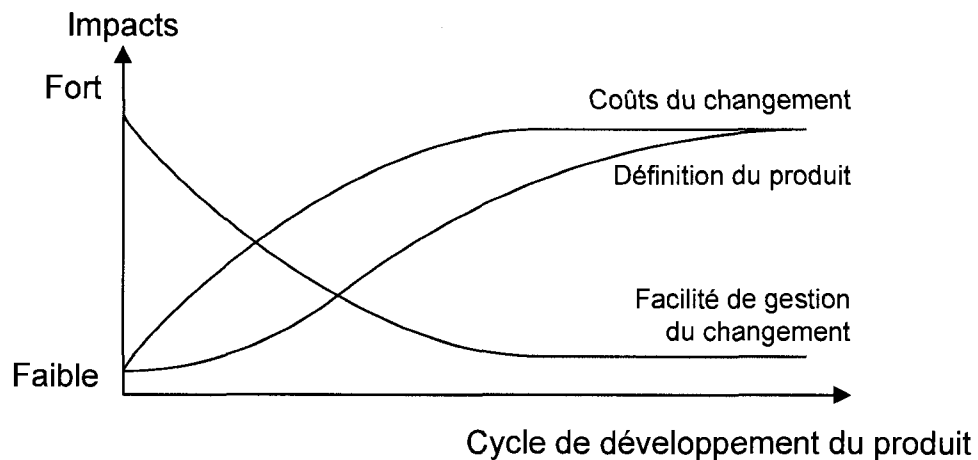


Figure 1.1 Impact du changement au cours du cycle de développement

Contrairement à l'ingénierie séquentielle, cette approche multidisciplinaire permet un chevauchement dans le temps des étapes du processus de conception (Figure 1.2), ce qui permet d'impliquer très tôt la fabrication dans le développement du produit.

Ingénierie séquentielle



Ingénierie simultanée

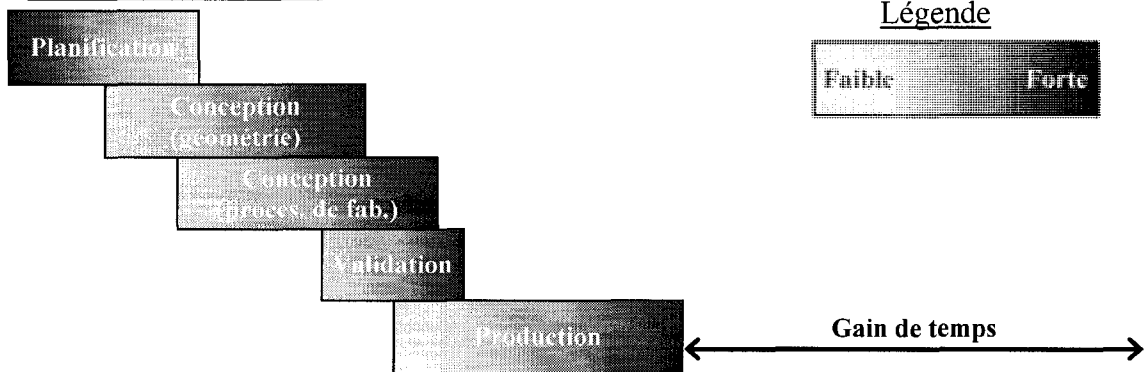


Figure 1.2 Principe de l'ingénierie simultanée, adapté de (Sky, 1996)

L'ingénierie simultanée permet donc une définition progressive du produit en impliquant très tôt la fabrication et l'assemblage. Elle implique en revanche une contrainte sur la définition du produit : l'impact du changement de conception du produit augmente considérablement avec l'avancement du cycle de développement (Figure 1.1) car il implique très vite de plus en plus d'intervenants (principe de l'ingénierie simultanée).

1.2 Conception de produit

1.2.1 Modélisation du produit

Depuis les débuts de la CAO, puis plus récemment avec l'apparition du développement de produit intégré, il est nécessaire de mettre en œuvre des modèles contenant plus d'informations que la simple géométrie du produit. C'est ce que l'on obtient avec le modèle de produit multi-vues ou la modélisation par *features*. Ces approches consistent à décrire l'objet en fonction de ses caractéristiques (fonctionnalité, fabrication, assemblage, recyclage, etc.) et non seulement de sa géométrie. D'après Gardan (Gardan, 2003), on différencie alors plusieurs types d'informations décrivant le modèle :

- Informations géométriques : B-Rep, CSG, etc.
- Informations technologiques : qui complètent les informations géométriques (perçage, filetage, taraudage, rainure, etc.).
- Informations de précision : qui définissent les tolérances de fabrication nécessaires au fonctionnement du produit.
- Informations matérielles : propriétés du matériau.
- Informations relatives au processus de développement : contraintes, assemblage, éléments finis, moyens de fabrication, etc.
- Informations administratives : référence, fournisseurs, stock, etc.

1.2.2 Vues en conception

La diversité des informations concernant le produit complique considérablement l'intégration des phases de son cycle de vie. Chaque intervenant désire voir les caractéristiques du produit propres à son domaine, c'est ainsi que l'on définit le concept de "vues". Anwer (Anwer, Mathieu, & Elmaraghy, 2005) distingue la vue géométrique, la vue conception, la vue fabrication et la vue contrôle métrologique (Figure 1.3) :

- La vue géométrique est la vue principale puisqu'elle est en relation avec toutes les autres vues.
- La vue conception correspond à l'activité du concepteur : élaborer le modèle en respectant les fonctions technologiques du produit.
- La vue fabrication est utile à l'ingénieur méthode, qui détermine la gamme d'usinage en fonction des surfaces fonctionnelles, des surfaces de posage, etc.
- La vue contrôle métrologique correspond à l'activité du métrologue, qui élabore la gamme de contrôle en fonction du produit, du processus de mesure et des spécifications géométriques.

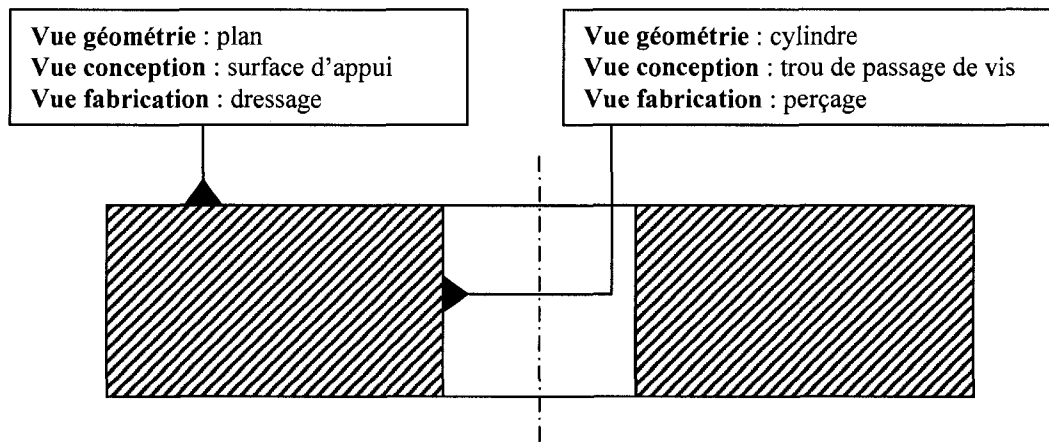


Figure 1.3 Les différentes vues d'un produit, adapté de (Tollenaere, 1998)

Il peut donc exister pour un même produit plusieurs modèles de *features*. Voici un exemple illustrant les différences d'interprétation du *feature* suivant le point de vue « métier ». La Figure 1.4 montre que suivant le point de vue de conception ou d'usinage, on considère les *features* différemment : le concepteur se place d'un point de vue fonctionnel, tandis que l'ingénieur méthode envisage la géométrie suivant les enlèvements de matière à effectuer à l'aide d'une machine outil.

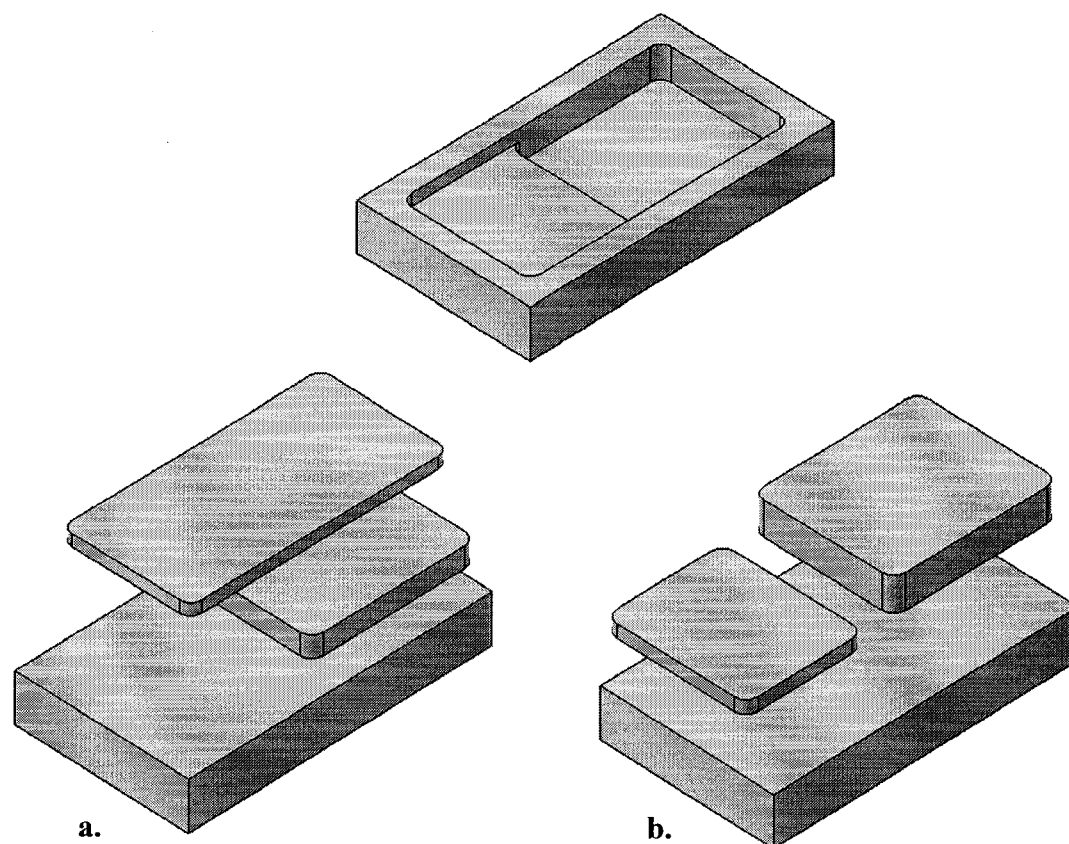


Figure 1.4 Interprétation d'un *feature* suivant le point de vue de conception (a) ou de fabrication (b), adapté de (Deneux, 2002)

1.2.3 Conception pour la fabrication

La conception pour la fabrication ("*Design For Manufacturing*" – DFM) a pour objectif de prendre en compte les diverses contraintes de fabrication (procédé, encombrement, usinabilité, paramètres machines limites, efforts de coupe, traitements, etc.) dès la conception de la géométrie de la pièce. Il s'agit en fait de rapprocher la conception du processus de fabrication afin d'optimiser les technologies, les procédés de fabrication et les coûts dès la phase de conception. Les objectifs sont généralement orientés vers l'assistance à la conception ou à sa semi-automatisation en prenant en compte les contraintes de fabrication (Tollenaere, 1998). On peut ainsi générer des gammes de

fabrication (automatique ou semi-automatique), modéliser le produit par des *features* d'usinage, ou générer des trajectoires d'outils pour des volumes standard (rainure, contournage, etc).

1.2.4 Conception pour l'assemblage

La conception pour l'assemblage ("*Design For Assembly*" – DFA) prend en compte des facteurs externes à la pièce comme : l'accessibilité, l'encombrement, la symétrie, l'orientation, le démontage, l'automatisation du montage, les coûts, la modularité, la facilité du remplacement pendant l'entretien, la présence de différents matériaux, etc.

Selon cette approche, les besoins fonctionnels du produit sont déterminés tout en optimisant le nombre et la géométrie des pièces, ainsi que les techniques utilisées. L'objectif final est de minimiser les coûts de production du produit. Différentes méthodes d'aide à la conception ont été développées pour l'industrie, comme celle de Boothroyd et Dewhurst (Boothroyd, Dewhurst, & Knight, 2002) qui ont réalisé une méthode très documentée et informatisée par la suite, concernant l'assemblage manuel et l'assemblage automatique. Ils proposent d'évaluer l'assemblabilité en codifiant la géométrie des composants et les opérations d'insertion en fonction de leur aptitude à être manipulés et insérés. Le résultat de l'analyse est un coût estimé de l'assemblage (efficience du produit) et une mise en évidence des paramètres de conception à optimiser.

Le principal inconvénient de cette méthode est qu'elle nécessite le dessin de définition du produit et sa séquence d'assemblage afin d'évaluer son paramètre d'efficience. On ne peut donc pas obtenir un retour sur information au cours du processus de conception. De plus, ces solutions (manuelles ou automatisées) requièrent de rentrer les paramètres de description des pièces indépendamment de la maquette numérique, ce qui implique une importante redondance des informations et une perte de temps. Les données numériques ne sont pas liées à la structure topologique, ce qui empêche toute interactivité avec l'utilisateur.

1.2.5 Conception pour X

Dans l'optique de l'ingénierie simultanée, la nécessité d'éviter les changements de conception tardifs (sources de surcoûts de développement et de production) a poussé le concepteur à chercher un retour d'information sur ses choix de conception dans de multiples domaines tout au long du cycle de vie du produit. Les méthodes DFMA pour la caractérisation de la fabrication et de l'assemblage présentées précédemment ont été les premières, mais elles ont été suivies par d'autres méthodes appliquées à différentes caractéristiques du cycle de vie (Billatos & Basaly, 1997; Kim, Lee, & Xirouchakis, 2007), regroupées sous le terme de conception pour X (ou conception pour l'industrialisation, "*Design For X*" – DFX) :

- le désassemblage (*Design for Disassembly*),
- la maintenance (*Design for Maintainability*),
- la fiabilité (*Design for Reliability*),
- la sécurité (*Design for Safety*),
- le facteur humain (*Design for Human Factors*),
- l'environnement (*Design for Environment*).

La méthode est utilisée à chaque fois afin de mettre en évidence les paramètres ou informations technologiques à optimiser pour mettre le produit en adéquation avec chaque étape du cycle de vie. On cherche en général (comme pour la méthode DFMA) à réduire les coûts et les délais de production.

1.3 Approche par feature

La notion de *feature* (que l'on retrouve également en français sous les termes d'entité ou de caractéristique métier) est présente dans de nombreux travaux sous les formes : *form feature*, *machining feature*, *assembly feature*, etc. Tous ces termes sont regroupés sous l'expression "*feature technology*".

Selon le groupe GAMA (Deneux, 2002), le *feature* est un moyen de communication naturel entre les intervenants (tout au long du cycle de conception et de vie du produit) et est fortement lié à l'application, au métier, voire à l'entreprise. En fonction de notre approche métier, on considère différemment le produit : seule la modélisation géométrique nominale est la même pour tous. Par exemple, l'alésage d'une pièce va être vue de différentes façons (Daniel Brissaud, 1992) par :

- le concepteur, comme caractérisant la cinématique du produit, avec des propriétés telles que l'état de surface ou la tolérance géométrique.
- l'ingénieur méthode, comme résultant d'une ou plusieurs opérations de fabrication (foret et alésoir), dont il a besoin de connaître la qualité requise par le concepteur.
- l'opérateur de fabrication, comme permettant le centrage de son montage d'usinage, dont il a besoin de connaître différentes propriétés afférentes à la résistance de la surface ou à la répétabilité du posage pour toute la série de pièces.

Les modèles développés actuellement dans les systèmes de CFAO sont essentiellement de type géométrique, leur description est donc limitée à la géométrie nominale (ce qui interdit une quelconque communication avec des contraintes de fabrication). C'est donc une nécessité de pouvoir gérer plusieurs types d'informations : géométriques, technologiques, de précision (état de surface, tolérancement), matériaux (propriétés physiques et mécaniques), calculatoire (éléments finis, commande numérique), administratives (référence fournisseur, stocks, délais), etc.

L'objectif principal de l'approche par *feature* est d'adapter les systèmes de conception à la démarche du concepteur.

1.3.1 Définitions

Le concept de *feature* a été introduit dans les années 1980 et à l'époque il associait la géométrie du produit à une opération de fabrication. Il a ensuite été beaucoup développé

par Shah, suivi par un grand nombre de chercheurs. Il est important de noter qu'il y a autant de définitions de *features* que d'approches de modélisation par *features*.

Le concept de *feature* a été évoqué pour la première fois par Grayer (Grayer, 1976) pour décrire la géométrie du produit en fonction de ses différentes opérations d'usinage. Chaque *feature* est donc une géométrie associée à une opération d'usinage de base (une rainure est obtenue par fraisage).

Une des définitions majeures a été proposée par Shah et Rogers (Shah & Rogers, 1988) : le *feature* est un ensemble d'informations relatives à la description de l'entité, qui a une signification technologique et possède des caractéristiques prévisibles. Cette définition largement reprise dans la littérature a été le point de départ d'une multitude de définitions dérivées.

Cunningham et Dixon (Cunningham & Dixon, 1988) définissent le *feature* comme une entité géométrique utilisée lors de la conception ou de la fabrication (ajustement, fonction, faisabilité, analyse des contacts, information sur les outils, utilité, etc.). Les *features* proviennent des connaissances ou des raisonnements utilisés lors de ces activités. Une autre interprétation de Dixon (Dixon, Cunningham, & Simmons, 1987) définit le *feature* comme une entité quelconque avec des attributs de forme ou de fonction.

La norme internationale ISO TC184/SC4 définit le *feature* géométrique comme une caractéristique individuelle : surface plane ou cylindrique, filetage, rainure, etc.

Plus récemment, le groupe de travail informel FEMEX (*Feature Modelling Expert*) a donné une définition pour rassembler la communauté de travail (Deneux, 2002). Le *feature* est donc :

- une unité d'information représentant une zone d'intérêt au sein d'un produit.
- décrit par les propriétés du produit (valeurs et relations).

- défini dans le cadre d'un certain point de vue sur le produit, relativement à une classe de propriétés et aux phases du cycle de vie.
- et peut être décrit par des propriétés issues de différentes classes de propriétés, ce qui permet de relier ces propriétés entre elles.

1.3.2 Représentation mathématique

Basée sur les définitions présentées précédemment, on peut proposer une définition mathématique du *feature* (Sy & Mascle, 2006) :

$$F_p = \{T, G, A_p, R\}$$

Le *feature* de produit F_p est constituée de toutes les caractéristiques nécessaires à son existence :

- Un pointeur T , qui le lie à la structure topologique de la pièce ou de l'assemblage.
- Une information géométrique G .
- Un attribut A_p , qui représente une ou des valeur(s) quantitative(s) caractérisant le *feature* F_p .
- Une relation R , qui lie le *feature* F_p aux autres *features* de produit.

1.3.3 Propriétés des features

L'avantage principal des *features* est qu'ils permettent de stocker une pluralité d'information, géométrique ou technologique, associée à de nombreux domaines. Msaaf (Msaaf, 2002) distingue deux types de propriétés :

- Propriétés intrinsèques :
 - La géométrie.
 - Le nom ou l'attribut de l'identifiant du *feature*.
 - Les caractéristiques du *feature* spécifiées par l'utilisateur.
 - Les tolérances intrinsèques (ex : tolérance de forme).

- Propriétés extrinsèques :
- Les dimensions et caractéristiques dépendant d'autres *features*.
- La localisation et l'orientation du *feature*.
- Les tolérances extrinsèques (ex : tolérance de battement, de position, d'orientation).

1.3.4 Utilité des features

Le concept de *feature* s'étend à des activités liées à la conception, la fabrication, la gestion de production, etc. Ainsi l'utilité des *features* est multiple et apparaît à chaque étape du processus de conception et de fabrication du produit :

- formaliser les connaissances du produit,
- collecter le savoir faire,
- améliorer la communication entre les intervenants tout au long du cycle de vie du produit,
- impliquer la fabrication très tôt dans le processus (déterminer les informations liées à la fabrication dès le début de la phase de conception) et ainsi réduire le temps de développement,
- codifier et mémoriser la sémantique d'une zone d'intérêt dans un modèle informatique (Deneux, 2002),
- favoriser la réutilisation (Deneux, 2002).

Les *features* permettent ainsi d'éviter les informations redondantes (principe du développement intégré) et de faciliter l'intégration avec les activités aval.

1.3.5 Utilisation des features

Il existe deux démarches de travail avec le *feature* :

- Conception par *feature* (*design by feature* ou "*Feature-Based Modeling*" - FBM), défini par Shetty et Lee (Lee, 1999; Shetty, 2002). La conception par *feature* est

utilisée actuellement par toutes les solutions informatiques de conception : le produit est modélisé à l'aide d'une bibliothèque de *features* disponibles dès le début de la conception (Figure 1.5). Chaque *feature* est construit à partir d'entités élémentaires (point, droite, face, etc) et a une fonction technologique. L'utilisateur agence les *features* dans l'arbre de spécification et donne les valeurs aux paramètres du *feature* (voir exemple sous CATIA™ V5 : Figure 1.6). Cette méthode a pour avantages de permettre la transmission d'informations "métier", et de faciliter les modifications (Tollenaere, 1998). Mais elle comporte selon Gardan (Gardan, 2003) deux inconvénients majeurs : elle impose au concepteur d'être pluri-compétent (notions de conception, de calcul et de fabrication), et elle limite la créativité du concepteur en imposant l'utilisation de *features* de fabrication (les *features* importantes en conception sont en général différentes de celles qui sont importantes en calcul ou en fabrication).

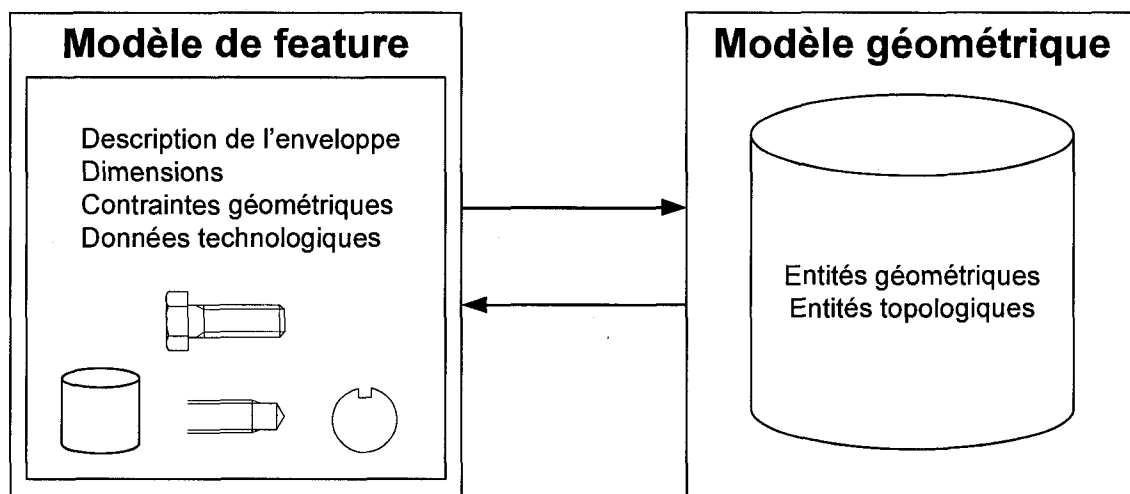


Figure 1.5 Modèle de *feature* associé aux caractéristiques géométriques, adapté de (Chang, Wang, & Wysk, 2006)

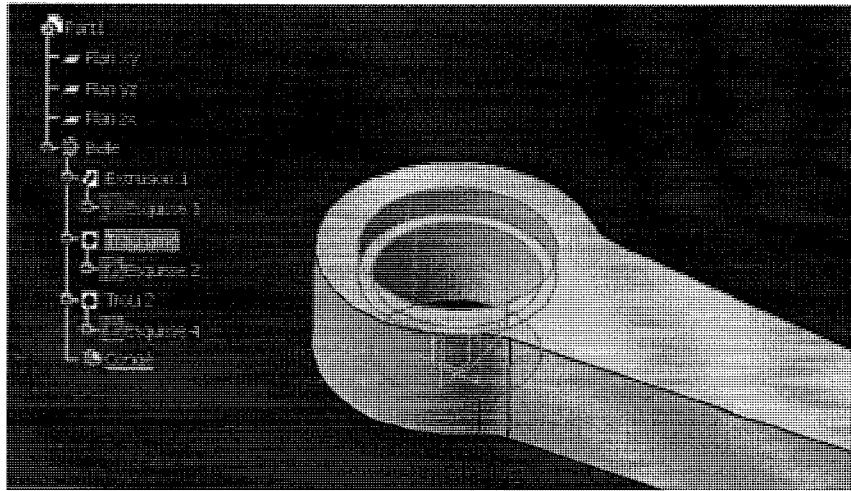


Figure 1.6 Exemple d'un arbre de spécification d'un produit modélisé par *feature*

- Reconnaissance de *feature* : avec cette méthode, on utilise un algorithme informatique pour détecter des *features* dans un produit. Cette reconnaissance dans le modèle géométrique peut aussi être assistée par l'opérateur (mode manuel, semi-automatique ou automatique). Mais l'état actuel des connaissances ne permet toujours pas de reconnaître automatiquement les *features* de tolérance ou d'état de surface, cette méthode est applicable uniquement aux *features* géométriques. D'après Shah (Shah, 1991), le système de reconnaissance de *feature* doit :
 - rechercher la structure de données,
 - extraire les *features* reconnus,
 - déterminer les paramètres des *features* (diamètre, profondeur),
 - permettre de compléter le modèle géométrique,
 - être capable de combiner les *features*.

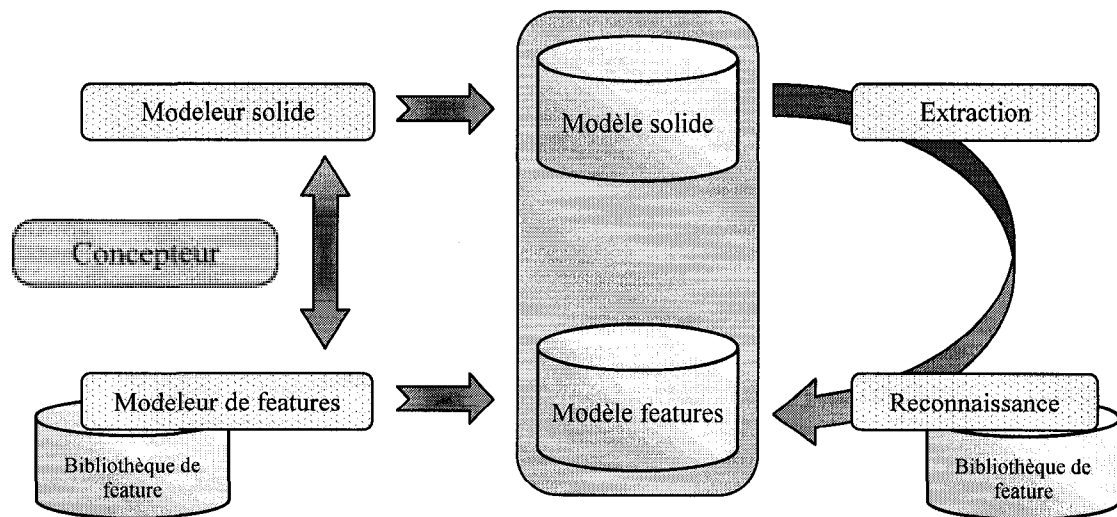


Figure 1.7 Mode d'utilisation des *features*, d'après (Shah, 1991) et (Deneux, 2002)

1.3.6 Les types de *features*

On a vu en introduction de ce chapitre que le *feature* est fortement lié à son application ou à son métier. On regroupe ainsi les différentes "vues" métier sous la notion de caractéristiques métier, car elles permettent de collecter un grand nombre d'informations utilisées tout au long du cycle du produit. Les *features* touchent à de nombreux domaines et peuvent être rassemblés en plusieurs groupes.

Cugini (Cugini, Mandorli, & Vicini, 1992) a déterminé deux types de *features* en fonction de leurs caractéristiques :

- *feature* de conception : c'est un *feature* de forme permettant de décrire une pièce mécanique et les fonctions de cette pièce.
- *feature* de fabrication : c'est un *feature* technologique lié aux outils de fabrication de la pièce (type d'usinage, précision, tolérances, etc.).

Selon Chep (Chep & Tricarico, 1999), la diversité des définitions de *features* dépend du nom du *feature* : le qualificatif que l'on place après le terme "*feature*" le détermine

(*feature* de forme, *feature* de précision, *feature* de fabrication, *feature* d'assemblage, etc.). Ainsi, la définition dépend du contexte. Dong (Dong, DeVries, & Wozny, 1991; Dong & Wozny, 1990) a référencé ces différents contextes :

- contexte de conception de produit (design) : les *features* sont des éléments de surface remplissant certaines fonctions (nervure pour rigidifier un composant).
- contexte de l'usinage (*machining*) : on considère les *features* comme des surfaces générées par des processus d'usinage connus (on peut obtenir un alésage avec un foret ou une fraise 2 tailles sur différents types de machines).

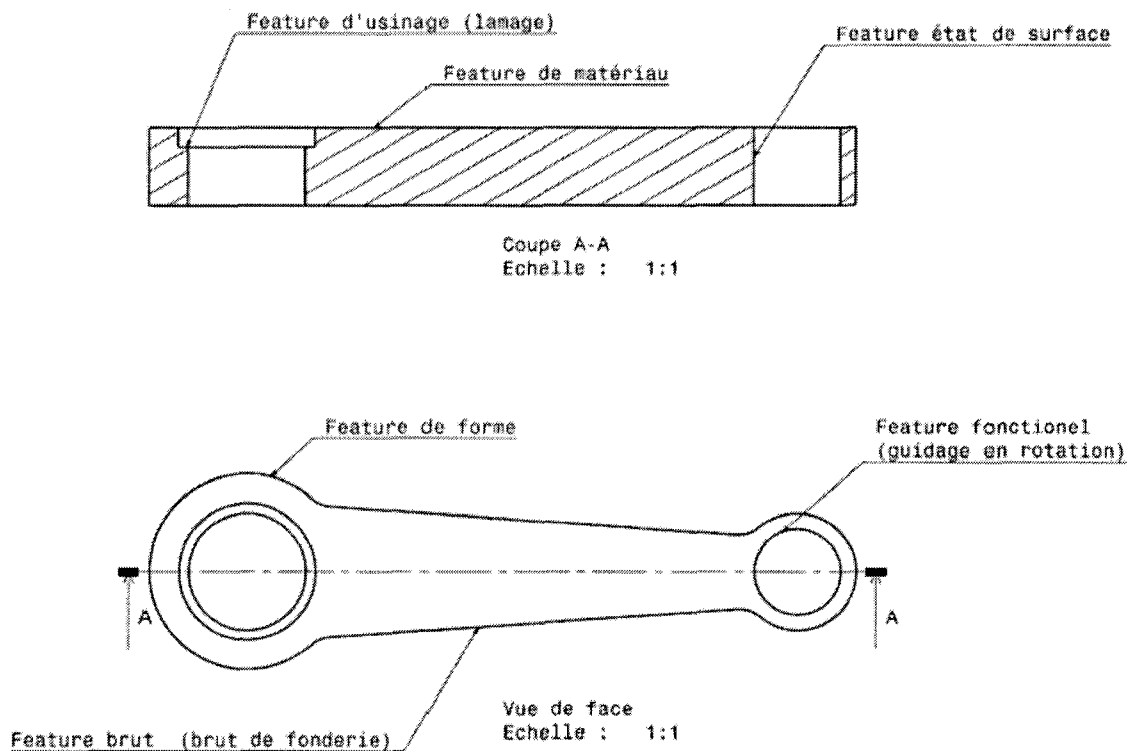


Figure 1.8 Exemples de *features* relatifs à une bielle, adapté de (Tollenaere, 1998)

Dans la littérature, on recense de nombreuses manières de classer les *features*. Chep (Chep & Tricarico, 1999) les range en quatre grandes catégories : les *features* de forme,

les *features* de précision, les *features* technologiques et les *features* de fabrication (Figure 1.8). Les trois premiers types sont appelés *features* primaires : le *feature* de fabrication est une composition des *features* primaires. Le groupe GAMA (Martin, 1999) classe les *features* selon qu'ils soient liés à l'activité de conception ou de fabrication :

1.3.6.1 Features liés à l'activité de conception

1.3.6.1.1 Feature de forme

Le *feature* de forme est un ensemble de faces représentant les frontières d'un objet. On distingue également les ensembles d'arêtes qui représentent les frontières du *feature* de forme (*boundary edges*). Ce *feature* exprime la géométrie nominale de l'objet.

1.3.6.1.2 Feature de liaison (ou abstrait)

Le *feature* de liaison peut être utilisé au cas où un ou plusieurs *features* ne peuvent être déterminés immédiatement dans le processus de conception (volume d'encombrement maximum d'un composant).

1.3.6.1.3 Feature topologique

Il définit les relations topologiques qui concernent les entités. Ces relations peuvent concerner des éléments d'un même *feature* ou de plusieurs *features* (Tollenaere, 1998).

1.3.6.1.4 Feature de précision

C'est l'ensemble de frontières qui ont des liens de tolérance (Chep & Tricarico, 1999). Selon De Jong (De Jong, Reitsema, & Hoogeboom, 1992) le *feature* de précision peut être divisé en *features* de tolérances et *feature* de rugosité (état de surface). Le *feature* de tolérance représente la dérivation acceptable de la géométrie nominale pour assurer la fonctionnalité, tandis que le *feature* de rugosité permet de lier le traitement de surface aux faces.

Shah (Shah, 1991) divise les *features* de tolérance en trois types, en se basant sur les principes du tolérancement géométrique :

- Le *feature* de tolérance associé à un paramètre du *feature* de forme. Il traduit la tolérance de forme (tolérance sur le diamètre d'un alésage ou sur la planéité d'une surface).
- Le *feature* de tolérance qui définit l'orientation entre différents éléments géométriques d'un *feature* de forme. Il traduit la tolérance d'orientation et le tolérancement géométrique (inclinaison, perpendicularité).
- Le *feature* de tolérance qui définit la position de chaque *feature* de forme par rapport aux autres. Il traduit la tolérance de position (localisation, coaxialité, symétrie).

Ainsi, le *feature* de tolérance peut être ajouté sous forme de paramètre à un *feature* de forme ou peut être défini comme un objet séparé qui concentre toutes les relations entre les *features* (Van Houten & Van 't Erve, 1992). On peut également classer les tolérances de battement ou les tolérances au maximum et au minimum de matière dans les différents types ci-dessus.

1.3.6.1.5 Feature de référence

Ces *features* sont essentiels à la viabilité du développement de produit intégré car ils définissent les éléments de référence de chaque composant (référentiel de référence, axes de référence, point de référence, plan de référence).

1.3.6.2 Features liés à l'activité de fabrication

1.3.6.2.1 Feature d'assemblage

Le *feature* d'assemblage est très utilisé par les derniers logiciels de CAO, il représente un point fondamental du développement de produit intégré et de l'ingénierie simultanée.

Une grande quantité de travaux concernent la modélisation d'un assemblage à l'aide de *features*.

Selon Shah (Shah, 1991) les *features* d'assemblage définissent l'orientation de la pièce, les surfaces d'interaction, les relations cinématiques.

Jabbour (Jabbour, 1997) a décrit une méthode de reconnaissance des caractéristiques de liaison d'un produit, qui permet d'établir un lien entre la géométrie et les procédés d'assemblage. Il a pour cela utilisé le principe des pseudo-matrices des demi-degrés de libertés de liaison introduit par Mascle (Mascle, 1990) pour décrire l'évolution de la structure du produit au cours de l'assemblage.

1.3.6.2.2 Feature d'usinage

Ce type de *feature* consiste en général à associer à une forme géométrique un processus d'usinage afin de générer automatiquement la gamme d'usinage et les programmes des machines (par reconnaissance algorithmique ou par conception directe avec des *features* d'usinage).

Selon le groupe GAMA (Tollenaere, 1998), certains attributs permettent de préciser la géométrie du *feature* et les caractéristiques du produit fini afin de définir le processus d'usinage adéquat :

- le type : il donne la pertinence des paramètres géométriques et les processus d'usinages envisageables.
- les directions d'usinage : elles donnent l'orientation de la pièce par rapport à l'orientation de la broche de la machine. On peut ensuite déterminer le volume occupé par l'outil tout au long de sa trajectoire à partir de la géométrie du *feature* d'usinage.
- la qualité de la surface : elle nous renseigne sur les caractéristiques de la surface (qualité, copeau minimum, etc.) à chaque étape d'usinage (ébauche, semi-finition, finition)

1.3.6.2.3 Feature de prise de pièce (mise et maintien en position)

Ce *feature* détermine les caractéristiques de mise en position et de maintien en position (informations géométriques sur les faces et sous-faces de posage et de bridage) et contient des informations technologiques (expérience métier) qui en assure la viabilité (D. Brissaud, Paris, & Tichkiewitch, 1997; Tollenaere, 1998).

1.3.6.3 Feature de base

1.3.6.3.1 Feature technologique

C'est un ensemble de surfaces frontières qui ont des liens technologiques entre elles. Les *features* technologiques définissent les caractéristiques de la pièce. Les auteurs en donnent une définition différente :

- Selon Chep (Chep & Tricarico, 1999), les caractéristiques concernées par le *feature* technologique sont : la composition du produit, les propriétés physiques et mécaniques (traitement thermique, type de traitement de surface : thermique, mécanique, chimique).
- Selon De Jong (De Jong, Reitsema, & Hoogeboom, 1992) ils regroupent les informations sur la gamme d'usinage, sur la maintenance ou les tests de performance. On les retrouve dans les applications de calcul en CAO (résistance statique, vibration, etc.).

1.3.6.3.2 Feature matériau

Le *feature* matériau a été introduit par Shah (Shah & Rogers, 1988), il contient les informations relatives à la composition des matériaux, leurs caractéristiques, à leur traitement, etc. Ils sont notamment utilisés dans les applications de calculs lors de la conception.

1.3.6.4 Feature administratif

Ils sont essentiels au concept de développement de produit intégré en permettant la gestion temporelle des pièces du produit. Ils contiennent des informations sur les références du produit, les fournisseurs, les stocks, les délais, etc.

1.3.7 Taxinomie des features

Comme nous l'avons vu lors de sa définition, il existe une infinité de *features*. On a donc cherché à les regrouper en classes, sous la forme d'une taxinomie. Le but poursuivi (Tollenaere, 1998) est de trouver des caractéristiques communes qui permettraient de limiter le nombre de supports de *features* : on aurait un support par famille de *feature*. Ce serait un avantage important dans la standardisation en vue de l'échange des données.

En reprenant les éléments de la littérature sur les *features* de tolérance, on peut créer la taxinomie correspondante (Figure 1.9). Duverlie (Duverlie, 1996) a également proposé une taxinomie des *features* d'usinage (Figure 1.10).

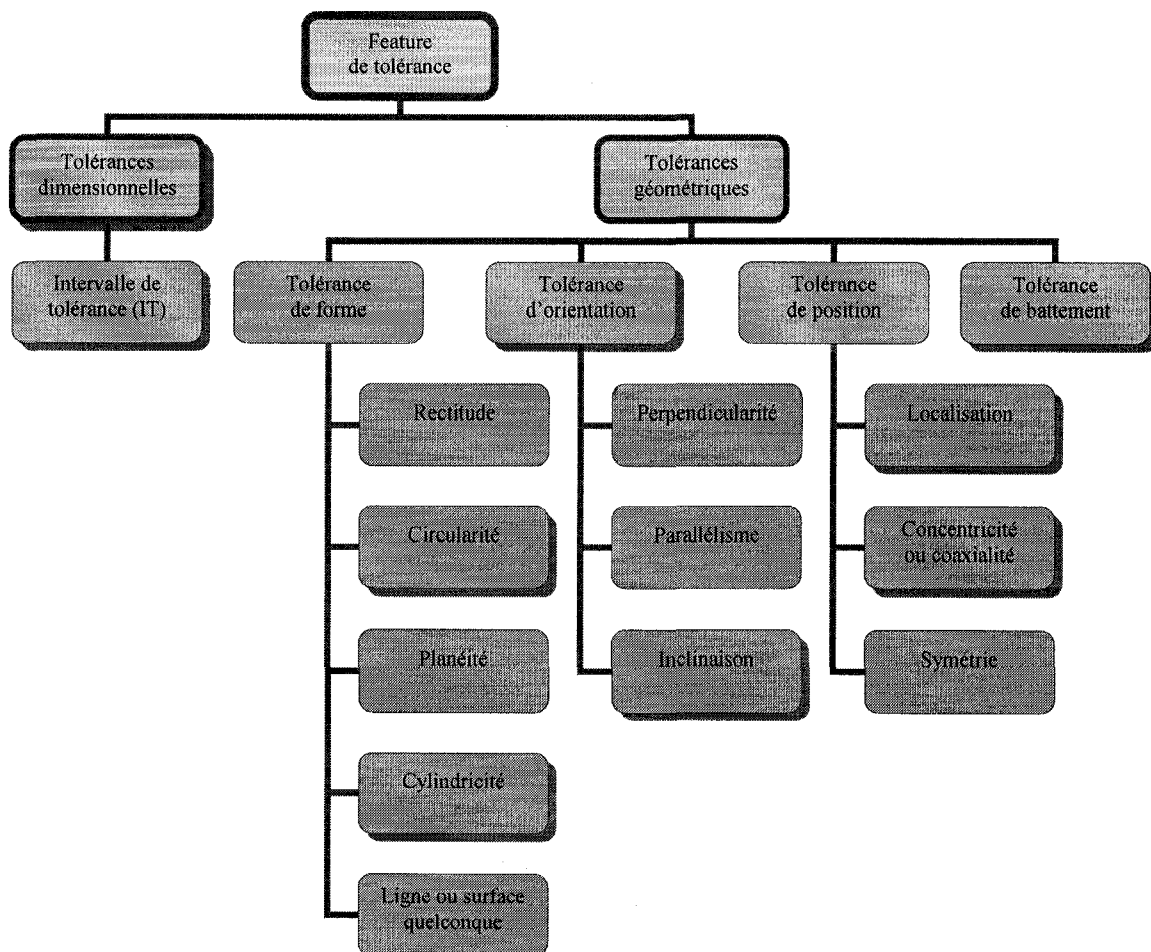


Figure 1.9 Taxinomie des *features* de tolérance, adapté de (Tollenaere, 1998)

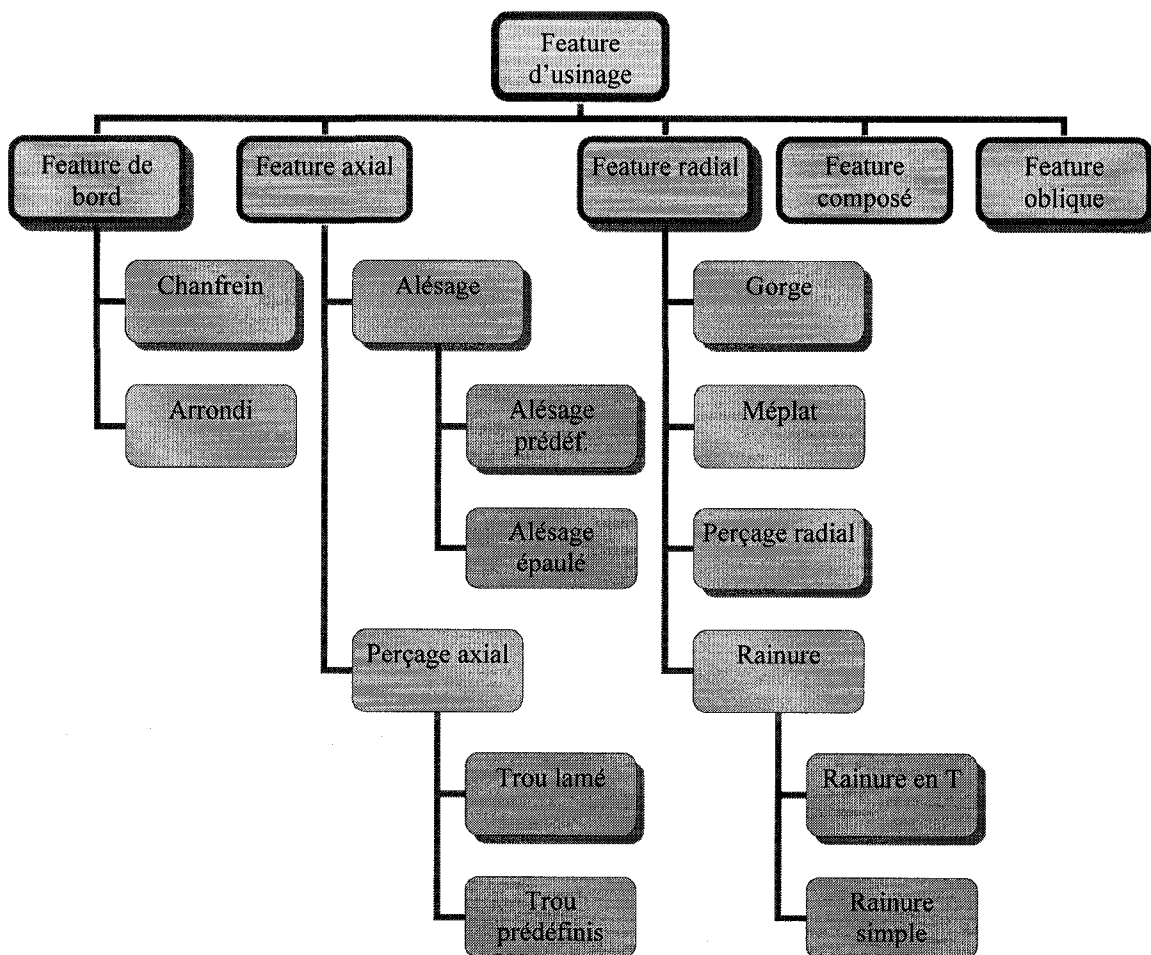


Figure 1.10 Taxinomie des *features* d'usinage en tournage, adapté de (Duverlie, 1996)

1.4 Conclusion

Dans cette revue de littérature ont été abordées les notions d'ingénierie collaborative et de conception de produit, qui couvrent les activités de recherche.

La phase de conception qui représente une très faible part du cycle de vie du produit est en revanche à l'origine de la plus grande partie des coûts. Dans un contexte de collaboration entre les différents domaines d'élaboration du produit, la circulation de l'information entre les intervenants et l'automatisation de certaines tâches prend

beaucoup d'importance. La représentation du produit doit être identique pour tous les acteurs du cycle de vie, alors qu'ils sont chacun concernés par des informations différentes.

Le modèle de *feature* apparaît donc comme le vecteur idéal de cette information technologique concernant le produit tout au long de son cycle de vie. Il permet de prendre en compte le domaine de chaque intervenant, tout en s'adaptant à la pluridisciplinarité des informations « métier ».

CHAPITRE 2 : SYNTHÈSE

Ce chapitre présente la démarche de l'ensemble du travail de recherche et l'organisation générale du document, basée sur les deux articles de revues scientifiques.

La première section présente les problématiques rencontrées au cours de l'étude du sujet et les limites que nous avons fixé. La deuxième section décrit les démarches et objectifs que nous avons suivis. Les sections suivantes présentent les différents modèles géométriques et la structure de données que nous avons choisis pour cette étude. La dernière section décrit le modèle utilisé pour la représentation et la construction de la structure de données.

2.1 Problématiques

2.1.1 Intégration de l'information

Durant les phases de conception préliminaire et d'élaboration du produit, les différents intervenants réfléchissent et mènent des études afin de prendre des décisions concernant la conception de la pièce. Les trois principaux paramètres qui sont alors discutés sont : forme, ajustement et fonction. Mais les acteurs de l'étape de la conception doivent aussi réfléchir à divers paramètres qui influenceront sur le cycle de fabrication et d'assemblage, et parfois sur le reste du cycle de vie du produit (ajustements pour faciliter la fabrication ou l'assemblage, dégagements pour permettre l'assemblage ou la prise de pièce, design des pièces pour faciliter le démontage et la maintenance, standardisation des pièces pour faciliter la maintenance et la gestion des pièces de rechange, facteur environnemental, etc). Tous ces concepts qui sont pourtant envisagés dès la conception, sont traduits dans

la géométrie du composant et restent implicites durant le reste du cycle de développement du produit.

Avec le concept de l'ingénierie collaborative, le département de conception transmet au département de fabrication l'information sur le produit sous forme de livrables. Pendant les stades suivants de l'élaboration du produit, les intervenants vont devoir déduire des données géométriques les informations nécessaires à leur tâche. Par exemple, afin de déterminer la chaîne de cotes, l'ingénieur des méthodes va analyser le produit pour trouver les surfaces fonctionnelles alors que cette information a déjà été traitée durant la conception. La maquette numérique ne garde pas de trace de ces données « métier », importantes pour l'élaboration des procédés de développement du produit. Toutes ces informations perdues doivent être re-déduite à partir de la structure géométrique dès qu'un utilisateur en a besoin, ce qui se traduit par une perte de temps et risque d'engendrer des erreurs d'interprétation, qui pourraient être évitées si l'on stockait l'information « métier » à chaque étape de la conception.

Les concepts d'ingénierie collaborative et de fabrication intégrée ("*Computer Integrated Manufacturing*" – CIM), de plus en plus utilisés en entreprise, sont basés sur le transfert d'information entre tous les acteurs de l'élaboration du produit. Pourtant, les représentations de données utilisées par les solutions actuelles de CAO ne permettent pas de représenter dans la structure topologique les données « métiers » comme les tolérances, les états de surfaces, etc.

Notre objectif est donc de permettre à chaque intervenant sur le produit de pouvoir « attacher » ces informations à la structure topologique du produit et de les faire transiter tout au long du cycle de vie.

En rendant l'information numériquement accessible, on ouvre la voie à un traitement automatique des données, une déduction automatique de certaines caractéristiques, une assistance aux calculs, etc. Nous avons vu dans le chapitre précédent que la méthode DFMA de Boothroyd était limitée par la saisie redondante de l'information et

l'extraction manuelle des paramètres nécessaires à l'analyse. L'intégration des données permettrait l'extraction automatique des paramètres requis à l'aide d'un *feature*. Des informations concernant les tolérances ou les états de surface peuvent nous permettre de générer automatiquement le dessin de définition, ou de définir les gammes de fabrication.

2.1.2 Aide à la conception

On a également vu dans le chapitre précédent que le processus de design est essentiel dans l'élaboration du produit. Les choix technologiques décidés par le concepteur influent directement sur tout le processus de fabrication et déterminent le temps et les coûts de production du produit. On estime que les décisions prises lors de la conception définissent entre 70 et 80% des coûts totaux de fabrication, et tout changement de conception ultérieur accroît considérablement les coûts de production. Il est donc essentiel, dans un objectif de réduction des coûts de définir le design définitif du produit le plus tôt possible.

En conséquence, chaque produit nécessite avant la mise en production une évaluation de divers paramètres qui devront être optimisés. Malheureusement, les éléments permettant d'optimiser ces choix sont rarement contrôlés par les concepteurs. Ils accomplissent donc leur tâche en utilisant leur expérience et en se basant sur la connaissance de l'entreprise : les produits développés par une même société dans le temps sont généralement du même type, une nouvelle génération est toujours basée sur un ancien produit avec des acquis et une connaissance « métier » très vaste, permettant subséquemment d'éviter bon nombre d'erreurs de conception.

Chaque décision du concepteur a un impact sur le reste du cycle de développement et parfois du cycle de vie. Il est donc indéniable que le concepteur a besoin d'un retour sur information de ses choix technologiques, afin de connaître l'effet de ses décisions sur les différents paramètres des processus de fabrication et d'assemblage. Une telle information lui permettrait de visualiser en temps réel les interactions entre les

différentes caractéristiques du produit et d'avoir un aperçu immédiat sur la pertinence de ses choix afin d'optimiser son design.

2.1.3 Limites

Les concepts présentés précédemment entraînent plusieurs interrogations quant à leur viabilité effective une fois intégrés à la méthode de conception. En effet, la méthode décrite ici est basée sur une meilleure intégration des données au début du processus de développement, ce qui implique inévitablement une augmentation de la charge de travail à l'encontre du concepteur (malgré les multiples avantages qu'elle lui procure). Le risque est que l'intégration des données devienne un frein à la conception en détournant l'attention de l'utilisateur ou en rendant sa tâche fastidieuse et par conséquent négligée. Il s'agit donc de déterminer l'information pertinente et le moment où la rentrer.

Une deuxième limite est également mise en avant, elle concerne la propagation des données avec l'évolution géométrique de la pièce. En effet, le concept d'ingénierie simultanée prévoit une évolution progressive du design du produit. Les informations attachées à une entité sont donc vouées à voyager au sein de la structure de données, il s'agit d'étudier la diffusion des informations en fonction des modifications de la géométrie.

2.2 Démarche et objectifs de recherche

L'objectif principal de ce mémoire est de concevoir un système d'assistance au concepteur, pour la prise en compte des contraintes « métiers », et capable de supporter l'analyse du processus de développement du produit. Il s'agit donc de développer un outil de CAO doté d'une interface utilisateur destinée à faciliter la saisie de l'information sur le produit. Le logiciel doit ensuite être capable d'extraire ces informations géométriques et technologiques de la structure de données, puis de les analyser afin de prévoir certains paramètres caractéristiques du processus de production. L'analyse des données doit également permettre une gestion des incompatibilités entre

les différents paramètres des entités (processus, procédés, caractéristiques, etc). L'objectif final est de mettre cette information à disposition de l'utilisateur par l'intermédiaire de l'interface CAO, pour lui permettre d'optimiser ses choix.

Le domaine couvert par le sujet étant extrêmement vaste, nous avons décidé de limiter l'étude du cycle de vie du produit du processus d'élaboration du brut, au processus d'assemblage.

La première étape du projet a été de créer une structure de données de produit capable de gérer la pluralité des informations concernant la géométrie de la pièce, mais également les informations technologiques, les relations entre entités, les informations fonctionnelles, etc. Cette structure de données est décrite dans le Chapitre 3. Les zones d'intérêt au sein du produit et de ses processus ont été représentées à l'aide de *features*. La structure de données a été implémentée sur la plateforme d'ingénierie logicielle OpenCASCADE (OpenCASCADE, 2007). Cet ensemble de bibliothèques open source programmées en C++ est orienté vers la CAO. OpenCASCADE nous a permis de combiner la base de données géométrique et la base de données technologique pour intégrer notre structure. Le logiciel créé est capable de faciliter la saisie des informations par l'utilisateur et de la centraliser grâce à une vue dédiée.

La seconde étape du projet présentée dans le chapitre 5, est axée sur l'aide à la conception en se basant sur l'utilisation de l'information technologique intégrée dans la structure de données développée lors de la première phase. L'extraction des données est également effectuée par des *features*. Les algorithmes de gestion des données ont été intégrés au précédent modèle logiciel afin d'offrir à l'utilisateur l'analyse des caractéristiques et de mettre en évidence les incompatibilités dues au choix de conception.

2.3 Définition des modèles géométriques

La structure de données que nous avons construite est basée sur la structure géométrique du produit. Avant de développer cette structure, il est nécessaire d'introduire les différentes modélisations géométriques principalement utilisées avec les logiciels de modélisation de produit.

2.3.1 Le modèle CSG

Le modèle CSG (*Constructive Solid Geometry*) est une structure de données qui décrit un solide par une succession d'opérations booléennes (Figure 2.1) sur des primitives élémentaires comme le pavé, le cylindre, la sphère, etc (Lee, 1999). Il a l'avantage d'avoir une structure simple à stocker, et donc à gérer. Il a également l'avantage, comme le modèle FBM d'être facilement paramétrable, ce qui est très utile dans notre étude. Mais ce concept de modélisation n'est plus utilisé par les logiciels de CAO actuels, il a été remplacé par le FBM, que nous avons décrit au chapitre précédent. Il reste cependant très utile lors de la manipulation des *features*, notamment pour l'extraction des données.

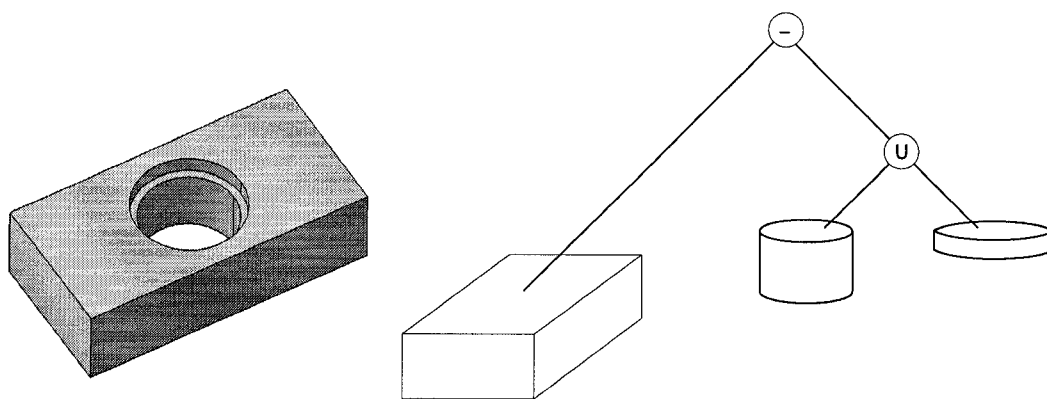


Figure 2.1 Arbre de modélisation CSG d'un solide (à droite)

2.3.2 Le modèle B-rep

Le modèle B-rep (*Boundary Representation*) est divisé en plusieurs types de représentation, mais elles sont toutes basées sur le même principe : les éléments qui

délimitent le produit (frontières) sont composés de faces, d'arrêtes et de points (voir Figure 2.3). Ces entités sont ensuite composées pour former une pièce. Cette structure de données a également l'avantage d'être très simple et très compacte.

2.3.3 Choix du modèle

Pour notre étude, nous avons sélectionné le modèle B-rep qui offre la possibilité de décomposer la pièce en entités élémentaires comme la face. En effet, la plupart des opérations de fabrication concernent la face sauf pour quelques procédés qui impliquent plusieurs entités ou la pièce entièrement (fonderie, déformation plastique, etc). L'entité *face* est donc le support privilégié pour véhiculer l'information tout au long du cycle de vie, comme le montre la Figure 2.2.

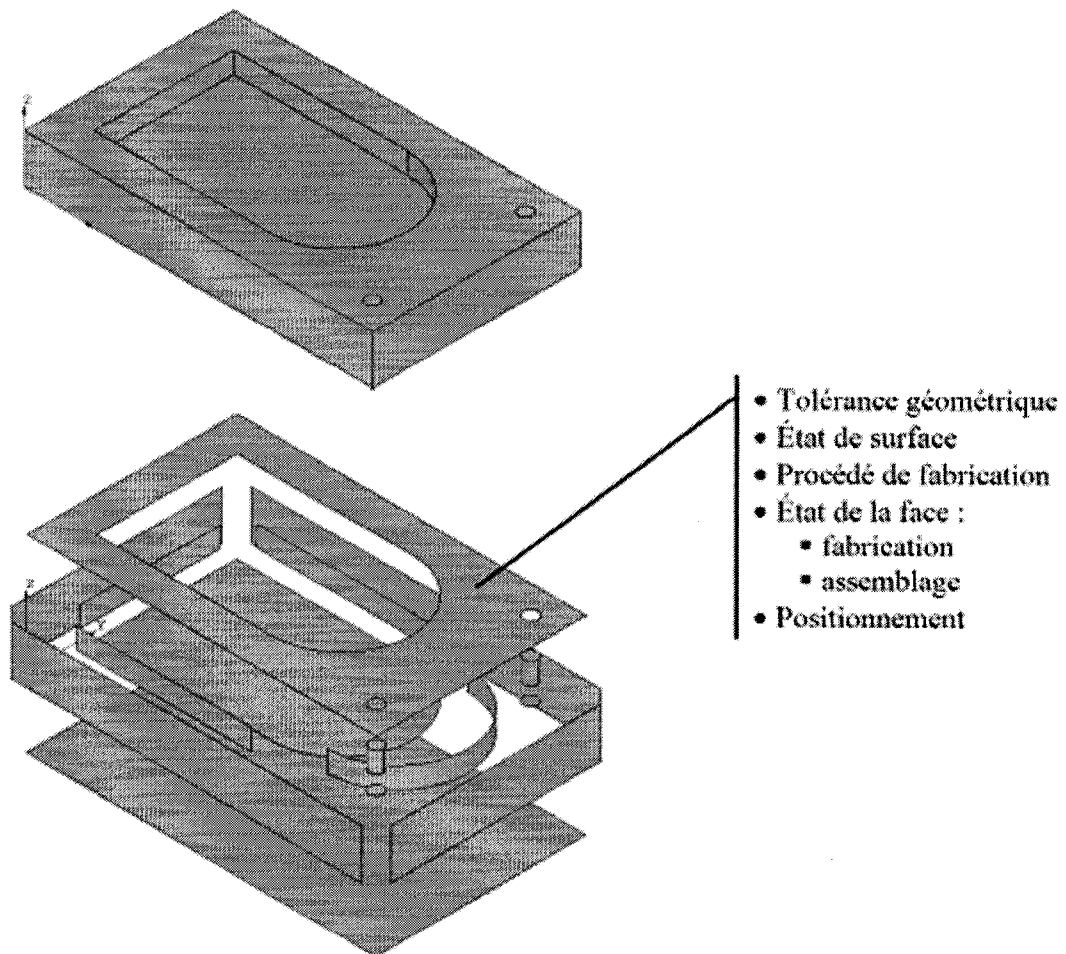


Figure 2.2 Intégration des données technologiques à la structure géométrique

Le modèle B-rep a été sélectionné pour sa facilité d'utilisation et sa bonne compatibilité avec l'information que nous désirons transmettre, mais notre étude pourrait également utiliser le model FBM plus couramment employé avec les logiciels contemporains. En effet, nous avons vu que le modèle FBM était également défini par ses frontières et basé sur les entités élémentaires du modèle B-rep.

2.4 L'organisation des données topologiques

La modélisation de la structure de données présentée dans le chapitre 3 est basée sur la structure des données topologiques. L'information véhiculée au cours du cycle de développement concerne en effet à chaque fois les entités géométriques du produit. Cette structure topologique est ici présentée en détail.

La topologie détermine le produit à partir d'entités géométriques élémentaires, agencées grâce à des relations (Paoluzzi & Pascucci, 2003). On peut en conséquence construire la géométrie d'un élément à partir de points (élément 0D), de courbes (élément 1D), de surfaces (élément 2D) et de volumes (élément 3D). L'entité la plus basique est le sommet (vertex). Le modèle B-rep permet ainsi de modéliser un objet tout en évitant la redondance d'entité dans la structure (Figure 2.3) :

- Le **sommet** (vertex) est une entité de dimension 0 : un point.
- L'**arête** (edge) est une entité de dimension 1. Une arête est constituée à ses deux extrémités de sommets.
- Le **contour** (wire) est constitué de plusieurs arêtes connectées par leurs sommets.
- La **face** (face) est une entité de dimension 2. Elle est délimitée par un contour externe fermé.
- L'**enveloppe** (shell) est constituée de plusieurs faces connectées par leurs arêtes.
- Le **solide** (solid) est une entité de dimension 3. Il est délimité par une frontière externe représentée par l'enveloppe.
- La **pièce** est générée par la fusion de solides.
- L'**assemblage** ou sous-assemblage, est une association de plusieurs pièces.

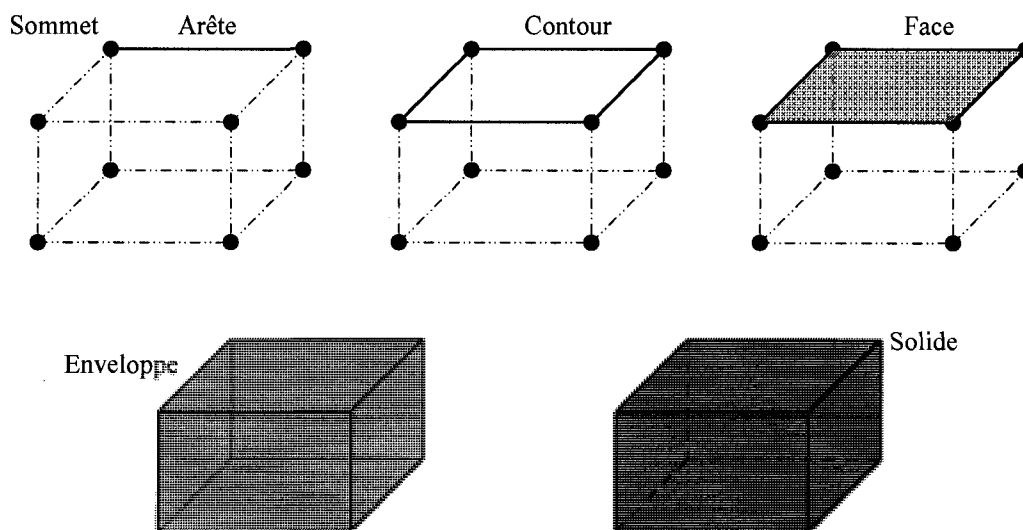


Figure 2.3 Topologie de la pièce (modélisation B-rep)

2.5 Le formalisme NIAM

Pour faciliter l'intégration des éléments topologiques et des données technologiques, nous avons choisi d'utiliser un modèle d'intégration orienté objet. La gestion des relations et des contraintes entre les entités nécessitait une méthode d'analyse statique, les aspects temporels et les états à l'origine n'étant pas abordés dans cette étude, les méthodes de représentation fonctionnelles ou dynamiques étaient inutiles.

Un simple graphe des liaisons entre objets et attributs n'est pas adapté à la représentation d'un produit. Nous avons opté pour une modélisation du rôle des objets ("*Object Role Modelling*" – ORM) pour construire la structure de données : le formalisme NIAM (*Nijssen Information Analysis Method*). Ce formalisme permet de représenter explicitement chaque entité du produit ainsi que de symboliser les différentes relations et contraintes entre ces objets.

Le formalisme NIAM est un outil d'analyse d'information qui facilite la création de la structure d'une base de données (Habrias, 1990; Halpin, 1995). Cette méthode a été normalisée en 1983 (TC97/SC5/WG3-N695).

On représente un objet ou un attribut respectivement par un cercle en trait plein ou discontinu. La relation qui lie chaque objet et/ou attribut est symbolisée par une boîte divisée en deux : le pont de dénomination (Figure 2.4).

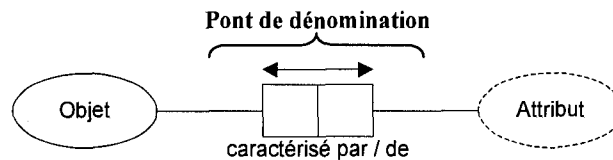


Figure 2.4 Représentation graphique du formalisme NIAM

Chaque case matérialise un lien unidirectionnel de l'entité la plus proche à l'entité la plus éloignée. Les contraintes d'unicité sont symbolisées par des doubles flèches au-dessus de l'une ou l'autre des cases : une flèche proche de la première entité (A) signifie que cette entité est en relation avec une seule et unique entité (B). La double flèche sur la totalité du pont de dénomination indique l'absence de telles contraintes, certains (A) sont en relation avec certains (B). On utilise la double flèche pour éviter les erreurs d'interprétation en cas d'oublis des liens d'unicité. La contrainte de totalité est exprimée par le quantificateur universel \forall , placé entre l'objet et le pont de dénomination.

Chaque type peut être subdivisé en un ensemble de sous-types, le symbole utilisé est la flèche unidirectionnelle.

On peut ensuite représenter des contraintes entre les différentes entités :

- la contrainte de totalité $\textcircled{\top}$, signifie que l'ensemble des sous-types représente la totalité du type parent.
- la disjonction $\textcircled{\times}$, signifie qu'un type ne peut se dériver que d'un unique sous-type.
- La disjonction totale $\textcircled{\top/\times}$ est une combinaison des deux contraintes précédentes.

- la contrainte d'unicité \textcircled{U} , signifie qu'un objet vérifie simultanément toutes les relations.
- la contrainte d'égalité $\textcircled{=}$

Le formalisme NIAM permet de transcrire des phrases en schémas relationnels. En voici un exemple en reprenant les symboles précédents (Figure 2.5).

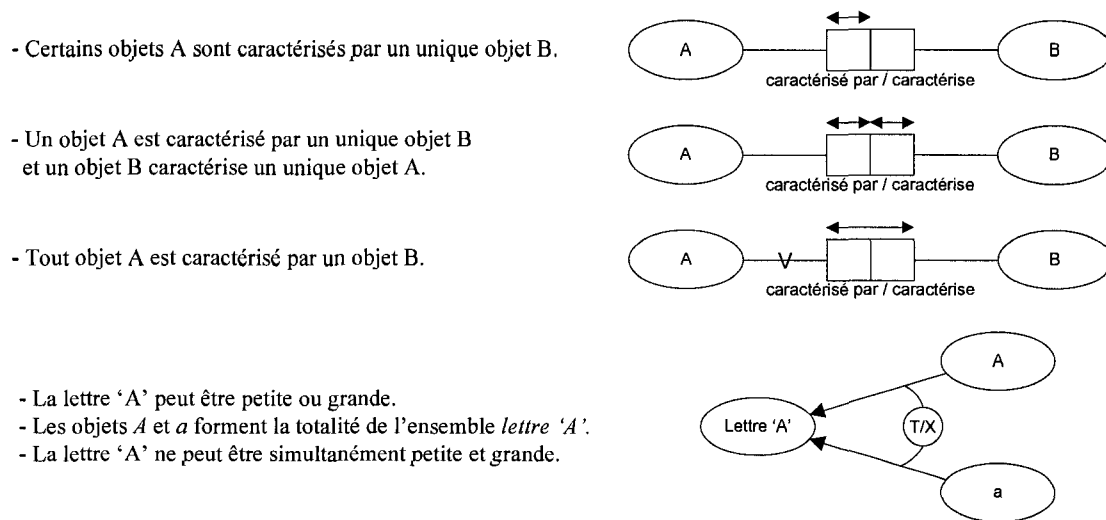


Figure 2.5 Exemples du formalisme NIAM

2.6 Conclusion

Ce travail de recherche nous a permis de répondre aux problématiques liées à la conception pour l'industrialisation. La flexibilité du formalisme NIAM nous a permis de représenter chaque *feature* dédié au cycle de vie du produit à travers la structure B-rep. Les structures de données ont été implémentées dans un logiciel basé sur la plateforme OpenCASCADE. Les chapitres suivants décrivent la construction de cette structure de données appliquée aux informations géométriques et aux processus de développement, ainsi que leur implémentation dans une solution logicielle capable de les exploiter.

CHAPITRE 3 : DATA STRUCTURE APPLIED TO PRODUCT LIFECYCLE

Soumis à *Research in Engineering Design*

Édition Springer

Ce chapitre présente la première partie du travail de recherche dont l'objectif était de développer un modèle de structure de données capable de supporter l'intégration des données techniques relatives aux produits, tout au long du cycle de vie. L'objectif à terme est de rendre ces données accessibles à tous les intervenants pour faciliter leur traitement et être capable de caractériser leur impact sur le cycle de vie du produit.

Les Annexes A et B reprennent la définition complète de l'entité face, développée au cours de cet article.

Abstract:

A great quantity of information is exploited during the product design, but only a small part of it is stored in the numerical model. This document suggests a data structure to remedy this problem which obstructs optimal product development. Our architecture is based on the topological information of a B-rep model and it deals with all information relating to the design, manufacture and assembly. It is built in order to be able to check the state of each entity at each step of the development cycle (state of manufacture of a face or state of assembly of a component) in order to generate directly the assembly or manufacturing data sheet. The software based on this data structure allows to pre-define

some characteristics according to the knowledge of the company or an other data base, to assist the various users. This approach allows to facilitate the data integration during the product development process.

Keywords:

Product design – Feature view - Design for X - Product feature - Product data.

3.1 Introduction

During the product development, the designer carries out a lot of studies which lead him to make decisions concerning the design of the part (adjustments to improve manufacture or assembly, maintenance, recycling, etc). However this information “translated” in the geometry of the component remains implicit throughout the lifecycle (from manufacture to maintenance and recycling). During the product development, the user will frequently have to deduce information which is necessary whereas these data already existed on the worksheets or the spirit of the various designers, but the digital mock-up did not keep any tracks. This information lost has to be re-found as soon as a user needs it: it leads an increase in waste of time and false interpretations which could be avoided if information was collected at each stage of the design.

The objective is to allow each user to “link” this information and to allow sending it throughout the lifecycle of the product.

This operation should not hamper the designer during his work, because the excessive increase in the workload is a barrier to the creativity. What might happen, if the designer is perturbed, is the negligence of this function. Then the software should select pertinent information and at which moment to enter them. The objective is to manage to automation of the task.

This study is based on the construction of a product data structure, which eases accessibility by the feature oriented user interface. NIAM formalism is used to build this structure.

3.1.1 Related works

3.1.1.1 Product modeling

Since the beginnings of CAD, and more recently with the appearance of the integrated product development (IPD), it is necessary to implement models containing more information than geometry data of the product: references, material and its properties, tolerance (Teissandier *and al.* 1999), assembly information (Lee *and al.* 1985; Noort *and al.* 2002), manufacturing information (Chang *and al.* 2006), etc. One obtains such results with the multi-view product model or feature modeling (Chep *and al.* 1999). These approaches consist in describing the object according to its own characteristics (functionality, manufacture, assembly, recycling, etc) and not only to its geometry. Gardan (Gardan 2003) differentiates several types of information describing the model:

- Geometrical information: B-rep, CSG, etc.
- Technological information: which complements geometrical information (drilling, threading, tapping, groove, etc).
- Precision information: which define the tolerances of manufacture necessary to the assembly operation and functionality of the product.
- Materials information: properties of materials.
- Development process information: constraints, assembly, finite elements, manufacturing, etc.
- Administrative information: references, suppliers, clients, stock, etc.

3.1.1.2 Different views of features

The diversity of information concerning the product complicates considerably the integration of the different lifecycle phases. The feature is a mean of exchanging

information. Each stakeholder wishes to work with the specific characteristics of his own field. That is how the concept of “view” is defined. Anwer (Anwer *and al.* 2005) distinguishes the geometrical view, the design view, the manufacture view and the dimensional inspection view:

- The geometrical view is the main view, that is why it is in relation with all the other views.
- The design view corresponds to the activity of the designers: to work out the model by fulfilling the technological function of the product.
- The manufacture view is used by the method engineer, who determines the manufacturing data sheet according to functional surfaces and set-up surfaces, etc.
- The inspection view is used to build the control data sheet according to the product, the process of measurement and the geometrical specifications.

The multi-view approach is very useful for this study because it allows to regroup diverse data in a single dedicated view. Using filtered and fitted information according stakeholder fields (feature view concept), the software facilitates their purpose.

3.1.1.3 Approach by feature

According to the GAMA group (Tollenaere 1998), the feature is a natural mean of communication between the different persons throughout the design lifecycle of the product. It is strongly related to the application, the trade or the company (technological or non-technological choices are currently reuse with the intention of decrease time and cost of development process). A user considers the product differently according to his trade, only the nominal geometric model (master model) is the same for all users.

Models currently developed in CAD and CAM systems are essentially of geometrical type, their description is thus limited to the nominal geometry (which prohibits any communication with constraints of manufacturing or assembly). In order to create a

“richer” model, one needs to manage several types of information: geometrical, technological, precision (surface quality, tolerance), materials (physical and mechanical properties), calculative (finite elements, numerical control), administrative (reference supplier and clients, stocks, time), etc.

The concept of feature was introduced for the first time by Grayer (Grayer 1976) to describe the product geometry according to its machining. A feature is defined as a geometry associated with a basic machining operation (a groove is obtained by milling, a hole by drilling). Further the concept was developed by Shah (1980), followed by various researchers. It is important to note that there are as many definitions of features as approaches of modeling by features. This first definition is very limited for our application because it concerns only geometrical information and not other data, which are essential for the lifecycle improvement.

Afterwards, one of the major definitions was proposed by Shah and Rogers (Shah *and al.* 1988): the feature is a set of information relating to the description of an entity, which has a technological meaning and foreseeable characteristics. This definition largely quoted in the literature inspired many derived definitions.

More recently, the informal working group FEMEX (Feature Modeling Expert) gave a definition to gather the community (Deneux 2002). According to this definition, the feature is:

- A unit of information representing a zone of interest within a product.
- Described by the properties of the product (values and relations).
- Defined according to a certain point of view on the product (the class of properties and the phases of the lifecycle).
- Described by properties resulting from various classes of properties, which allows to link them.

This description jibes with our study: it allows the description of each entity of the product, to attach various data to the geometry, to manage different views, etc. The feature concept is used from the beginning to the end of the product development process. It lets to format knowledge of the product and improves the communication between persons involved throughout the lifecycle of the product. In addition, this concept supports the information re-using, during or after the product design process. Feature also allows to avoid redundant information (this is one of the main principle of the integrated product development) and to facilitate integration with the downstream activities. The main objective of the feature approach is to adapt the systems of design to the method of the designers.

3.1.2 NIAM Formalism

3.1.2.1 Potentials

This static analysis method allows the management of the data structure (the functional or dynamic methods allow to show the system operation, to analyse the temporal aspects and to study the initial state, etc). It is focused on the relations and the constraints between the data of the modelled system; it is thus optimal for the development of a data model.

3.1.2.2 Description

NIAM Formalism (Nijssen Information Analysis Method, generalised recently as Natural language Information Analysis Method) is a tool for information analysis which facilitates the creation of a data base structure (Halpin 1995): it aims at describing the system in order to reproduce it. This method was standardized in 1983 (TC97/SC5/WG3-N695).

Objects and attributes are represented respectively by full and dashed ellipses. Links between each object and/or attribute are represented by adjacent boxes containing the

relationship names: the designation bridge (Figure 3.1), which is connected to the entities by full lines.

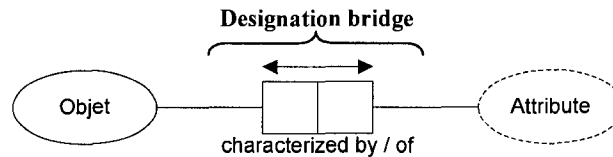


Figure 3.1 Schematical representation of NIAM formalism

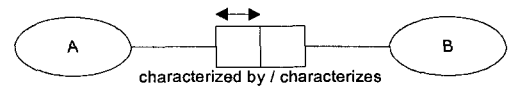
Each box represents a unidirectional link between the nearest entity to the most distant entity. Uniqueness constraints are symbolized by double arrows above both of the boxes (the double arrow on the totality of the designation shows the absence of such constraints). The constraint of totality is represented by the universal quantifier \forall , between the object and the designation bridge. Each type (object) can be subdivided in a set of sub-types. In this case, the symbol used is the unidirectional arrow.

One can represent constraints between various entities:

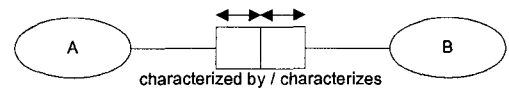
- The totality constraint $\textcircled{\top}$, means that the whole of the sub-types represents the totality of the relative type.
- The exclusion constraint $\textcircled{\times}$, means that only one sub-type can be derived from a type.
- The totality exclusion constraint $\textcircled{\top/\times}$ is a combination of the two previous constraints.
- The uniqueness constraint $\textcircled{\cup}$, means that object checks simultaneously all the relations.
- The equality constraint $\textcircled{=}$.

NIAM formalism allows to transcribe sentences in diagrams. The following example resumes the previous symbols (Figure 3.2).

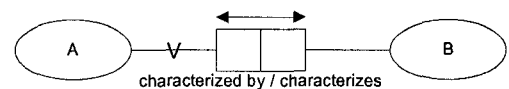
- Some objects A are characterized by an unique object B.



- An object A is characterized by an unique object B and an object B characterizes an unique object A.



- Every object A is characterized by an object B.



- The letter 'A' can be an uppercase or a lowercase.
- The objects *A* and *a* represent the totality of the assembly letter 'A'.
- The letter 'A' can't be simultaneously an uppercase and a lowercase.

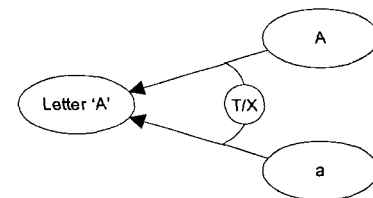


Figure 3.2 Examples of NIAM formalism

3.2 Modeling of the product lifecycle

3.2.1 The manufacturing process

The manufacturing process is a succession of stages which will gradually modify the part from the raw state to the final state. For any stage of manufacture (foundry, forging mill, machining, assembly), one can define the state of each entity of the topological structure. The different states of each entity *face* of the product are gathered in the part state graph. The part state graph is focused on the face state because the face entity is generally linked to each operation of manufacture. But this modeling is not very representative for specific processes like foundry or forging mill, where the whole part can be generated in only one stage.

Information concerning the manufacturing process has to be directly linked to the part and the face state (Figure 3.3). One can thus decompose all the stages and analyze the changes face by face via the state graph. This graph allows a direct access to information

necessary to the generation of manufacturing data sheet. Indeed, the successive states of each entity *face* correspond tot the various stages of the manufacturing process.

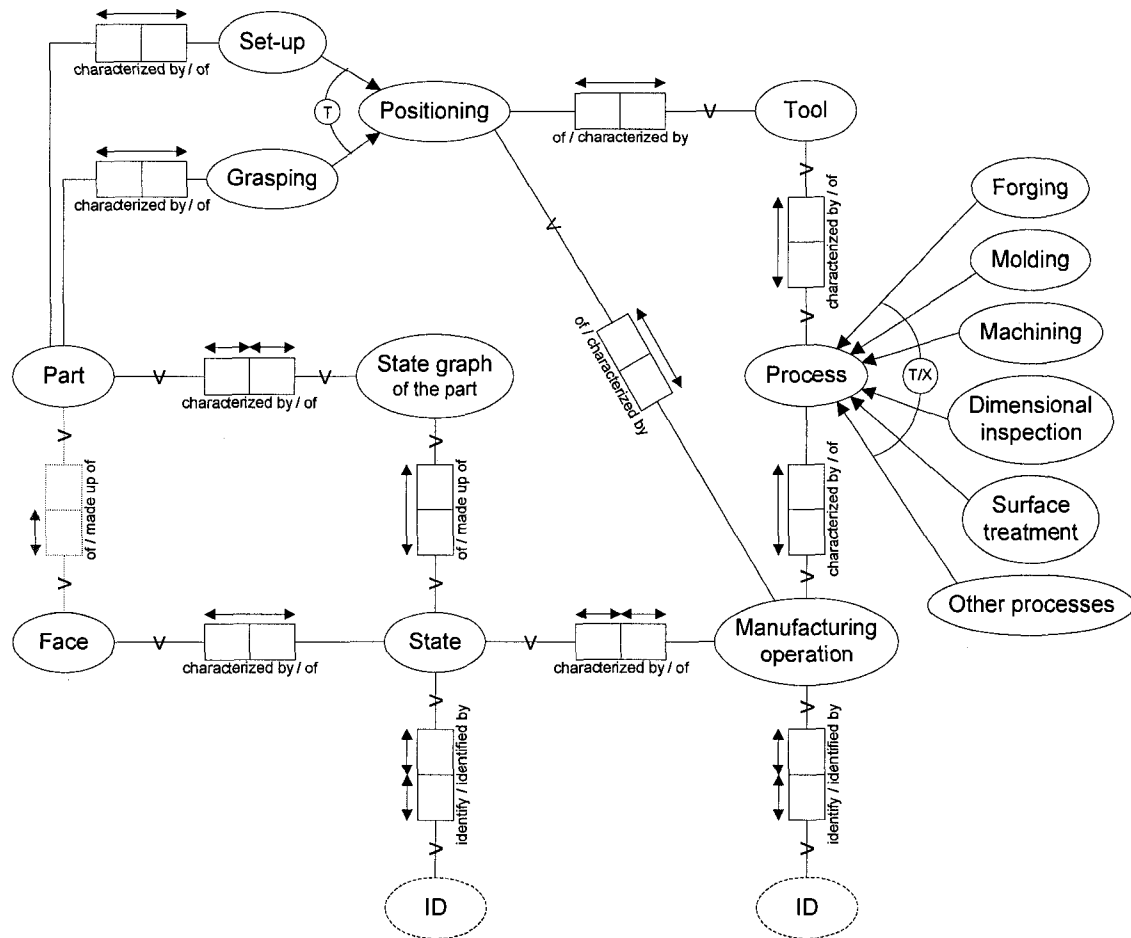


Figure 3.3 Schematical representation of the manufacturing process

Each state is associated with a manufacturing operation and this operation is linked with a process. One can divide the object *process* into five main sub-objects: molding, forging, machining, surface treatment and dimensional inspection. Each process sub-object can be expanded, divided into a lot of objects and attributes, in a feature taxonomy graph (Tollenaere 1998). The Figure 3.4 shows a partial example of the manufacturing sub-object (but such development involves sizeable graph). Each child sub-object is also divided in elemental object.

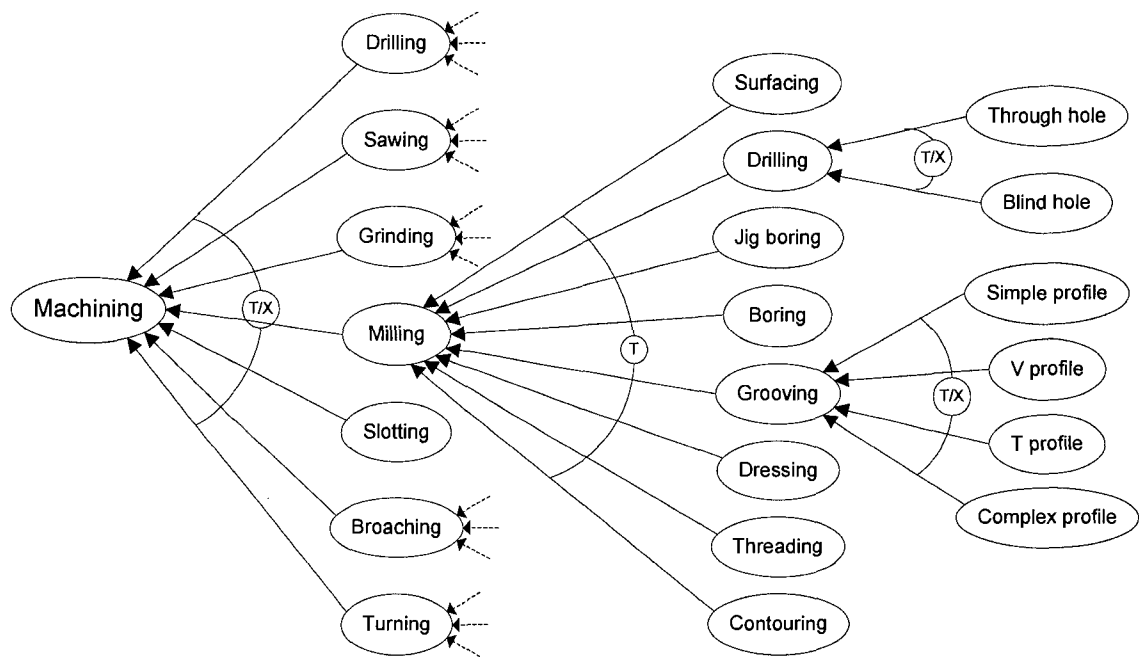


Figure 3.4 Partial expanding of a process sub-object

The complete process management (associated to the tools) by the user is very restricting. It can lead to an overload and the disinterest of designers. An automation using features discharges the users. Indeed, one can define standard processes with main preset characteristics according to some types of parts and their applications. The most interesting is to preset features according to the know-how and capability of the company: old manufactured solutions, machining equipment, etc. This stage can be carried out manually using a graphical user interface, or by a semi-automatic selection from a data base of similar parts.

Information concerning the positioning of the part relies on the machine, deduced from the process, the geometrical tolerances, the dimensional tolerances and from the topological data of the part. The selection of these data can be facilitated using features, the user is request only for some information. One can also automate the selection like the determining of the process parameters (the intervention of the user is limited).

This model is well-adapted to describe sequential operations of the product development like machining (face by face changes) or surface treatment (face by face or product changes). These processes lead a succession of stages which gradually modifies the product entity by entity and are well gathered in the data structure. Entities are updated by a basic manufacturing feature (Miao *and al.* 2002). In the case of molding or forging mill, many entities of the product are created or modified during the same stage. That is why the related manufacturing feature should be linked to the process of entity in the Figure 3.3 and also to the process of other concerned entities. Each entity is affected by various factors according to the process and the global topology of the part: risers, denseners and irregular shrinkages according to the geometry, various machining allowances according to technological functions and following processes. These parameters are set by the manufacturing feature.

3.2.2 The assembly process

The manufacturing process and the assembly process are very similar: the assembly process is also a succession of stages which will gradually modify the product by the addition of parts. It is thus possible to define the state of the assembly (every component) at each stage of the process. One can introduce by analogy the state graph of the product (Figure 3.5), which gathers the states of each element (entity part) of the product. In the case of the assembly process (contrary of the manufacturing process), the states graph of the product is focused on the state of the parts (more particularly of the face) because these entities are affected at each stage of the assembly. Sometimes, the assembly process needs to be defined according to various other parameters like geometrical tolerance, stability, accessibility, parallel assembly, etc. The dialog between these objects and there attributes is assured by dedicated features.

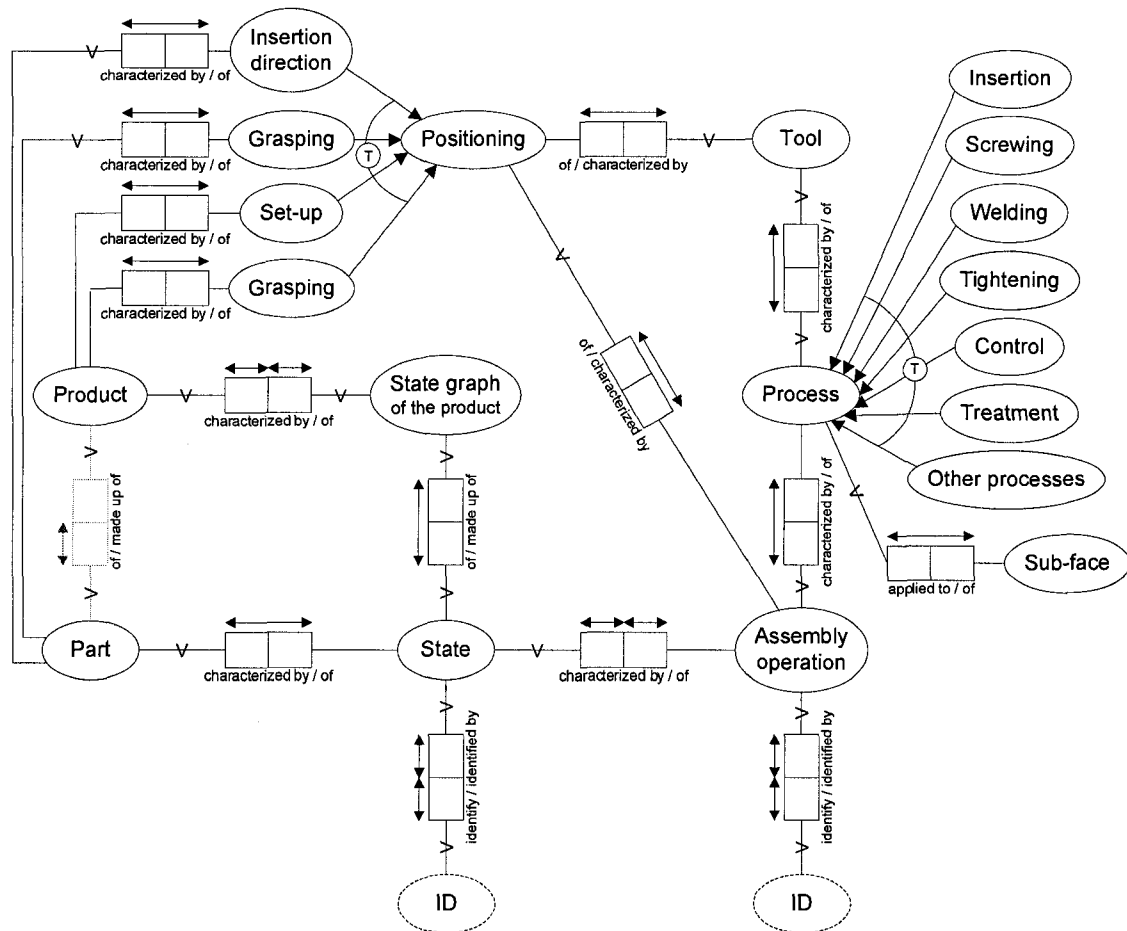


Figure 3.5 Schematical representation of the assembly process

The object *product state graph* is linked directly to the assembly and the state of the part. Such data arrangement allows a step by step analysis of the changes which gives a decomposition of the assembly operation, directly linked to the assembly data sheet.

Each state is associated to an assembly operation, itself linked to a process and a positioning system. The object *process* is sub-divided in many sub-objects: assembly process, control, treatment. This object is also related to a *sub-face* object (Jabbour and al. 1998), introduced by the necessity to model some parts of the face making inaccessible by the assembly operation.

The positioning system is defined by characteristics depending on tools and selected process. The mobility of parts relating to the product requires to differentiate the positioning. The set-up on the machine tool is linked to the product entity with the grasping. These parameters are directly related to the geometry and the functional surfaces. Grasping and insertion direction of the part are also linked to these parameters. The assembly features require also a pre-positioning object linked to the part.

3.3 The organization of the topological data

During the lifecycle, information can link to a multitude of entities of the product (assembly, sub-assembly, part, face, etc) and can share out a multitude of fields (tolerance, assembly data, treatment, materials, maintenance, recycling, etc). The main objective is to determine the way of processing this information and to understand the classification of topological data of the product.

3.3.1 Topology of the part

Topology determines the product from elementary geometrical entities, arranged with relations (Paoluzzi *and al.* 2003). The geometry of the product is build from four main entities: vertices (0D element), edges (1D element), faces (2D element) and solids (3D element). The most basic entity is the vertex, at the root of geometrical data. The B-rep model makes object modeling possible while avoiding the redundancy of entity in the structure:

- The vertex is a 0 dimensional entity, a point.
- The edge is a 1 dimensional entity. An edge is made up at its two ends of vertex.
- The wire is built from several edges connected by their vertex.
- The face is a 2 dimensional entity. It is delimited by a closed external boundary. This entity is connected to the part state graph (Figure 3.3).
- The shell is built from several faces connected by their edges.

- The solid is a 3 dimensional entity. It is delimited by a closed external boundary materialized by the shell.
- The part is generated by assembly of several solids. This entity is also connected to the product states graph (Figure 3.5).
- The assembly or sub-assembly is made up of several parts.

OpenCASCADE uses this decomposition of the part's topology, but each geometrical modeller has his own topological data structure. This study is based on the topology of the part used by OpenCascade to build the data structure of the product.

3.3.2 Construction of the topological structure

For a functional data transfer through the lifecycle, information must be associable with any entity (edge, face, part, assembly, etc). That is why all data are added to the primary tree (Figure 3.6). The topological structure is the skeleton of the information exchange: it allows to link product modeling to the various technological features.

Each element of the topology is identified by a single attribute (identification number). All additional information about the product are linked to the right-hand branch of the tree.

Entity *part* is characterized by an identification number and a name, but also by an extrinsic feature like material. The Figure 3.7 shows a schématical representation of the material feature defined by its name, its designation, its chemical composition and its strength properties (both are automatically deduced from the designation, but strength properties can be modified by various treatments).

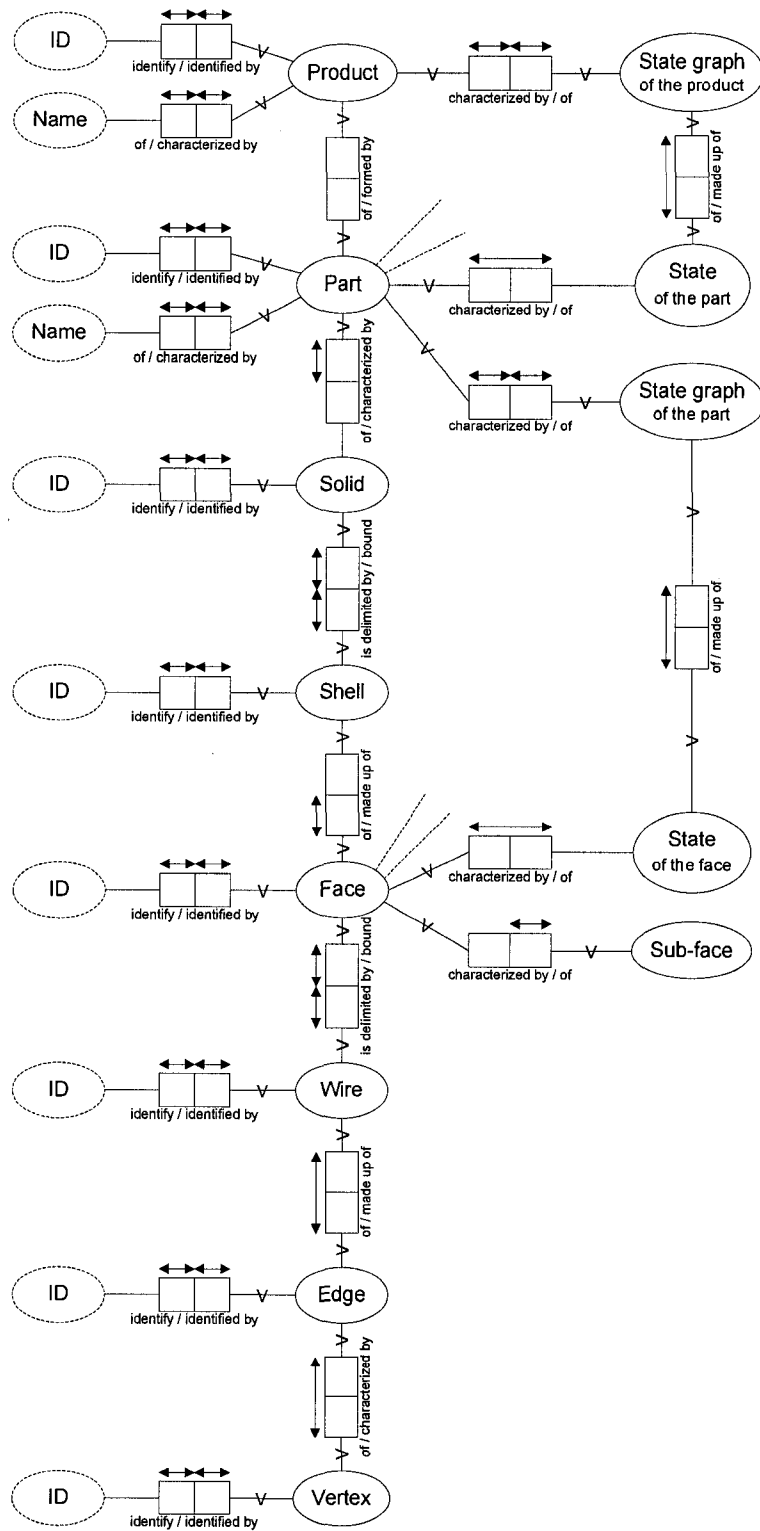


Figure 3.6 Schematic representation of the product topological structure

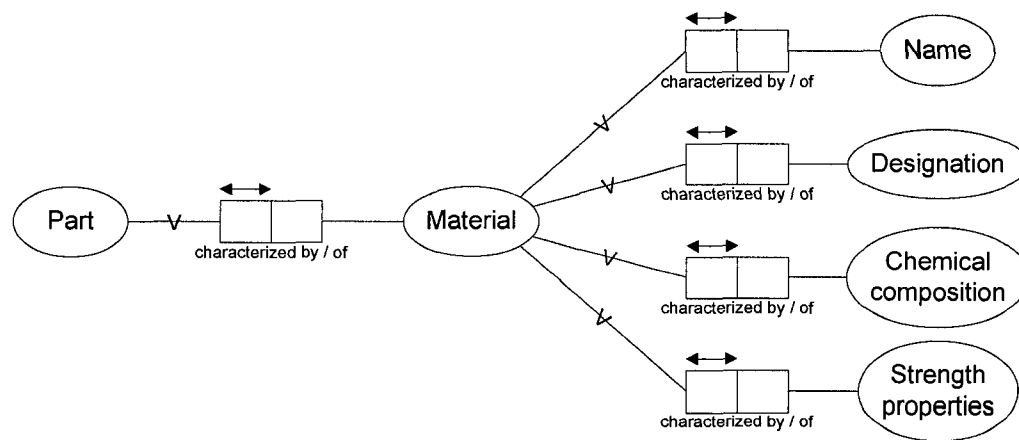


Figure 3.7 Schematical representation of the material feature

The great part of information which is necessary to save during the design is related to the entity *face* (tolerance, roughness, machining state, etc) and sometimes could be generalized to the entity *part* (surface treatment, machining and assembly state, etc). One can gather the majority of this information in two main categories: intrinsic and extrinsic characteristics of the face.

Among the properties of the face, one finds: information relating to geometrical tolerances, dimensional tolerances, the surface quality, the surface treatments, and the state of the face (for manufacture, assembly, etc).

Some data of these features depend on intrinsic and extrinsic characteristics, they can thus be automatically inferred from the topological structure or from the other features. The methodology differs according to the type of data. The feature recognition (Dimov *and al.* 2007; Dong *and al.* 1997), allows to detect form features (Coma *and al.* 2003) or manufacturing features (Gao *and al.* 1998; Regli *and al.* 1995) in the part topology from a feature library. Some attributes can be set or pre-set by this recognition: support element of the datum, type of tolerance, etc. Rules between features are sometimes required and could facilitate the use of stakeholders. One can thus pre-define authorized manufacturing process, type of tolerance, face state, required machining surface

(essentially according to technological function information and roughness value), etc. An other manufacturing feature can be applied to the tolerancing to set technological information. Indeed, datum entry order shows up precedence constraints, and this information is useful for the manufacturing data sheet generation (selection of machined surfaces, choice of set-up surfaces, etc).

3.3.2.1 Geometrical tolerance

Entity *face* is linked to the geometrical tolerance (Figure 3.8), but it is also connected to the datum. A form feature analysis allows to extract the support element from the datum and to pre-set data about the toleranced entity. The tolerance zone depends directly on the value of the tolerance, the concerned element (the face) and a possible modifier. The *geometrical tolerance* type is divided into three sub-types: location tolerance, orientation tolerance and form tolerance. The taxonomy of the geometrical tolerance is shown at the right of the Figure 3.8, it defines automatically the tolerance zone with the missing parameters. To optimize the information saving by the user and to allow the generation of different tolerance zones, a feature must be able to manage each sub-type.

According to the ISO standard 1101 (Quatremer 1996), the toleranced elements can be a surface or an edge. This part of the data structure that worked out for the entity *face* can also be applied to the entity *edge*.

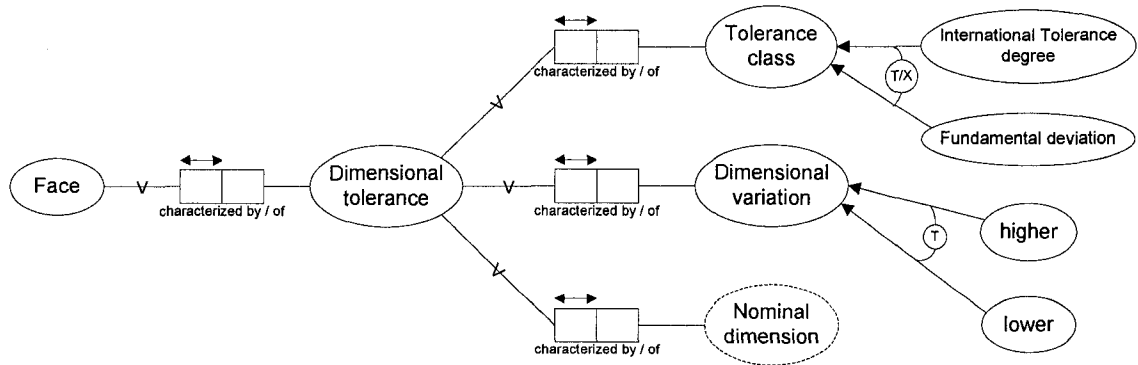


Figure 3.9 Schematical representation of the dimensional tolerance feature

3.3.2.3 Face state during the manufacturing process

This feature allows to know the state of the face throughout the manufacturing process of the product (molding, machining, treatment, dimensional measurement, etc). It is directly related to the state graph, this object thus appears in Figure 3.3. This feature is updated by information coming from manufacturing operation and process objects.

3.3.2.4 Surface treatment

The surface treatment object (Figure 3.10) is characterized by a process (which appears on the Figure 3.3, it is also a sub-object of the *process* object), a thickness (or depth according to the treatment) with its own tolerance, and the type of deposited material. The majority of the necessary information can be gathered in a preliminary data base, which facilitates the settings by the users. A feature can also manage the constraints imposed by the different choices. The objective is to inform the designer about the consequences of each option: the material of the product limits some choices, the type of process involves different deposited material choices, etc. The partial taxonomy of the surface treatment feature represented in the Figure 3.10 can be expanded for all other sub-object (conversion, mechanical and coating treatment).

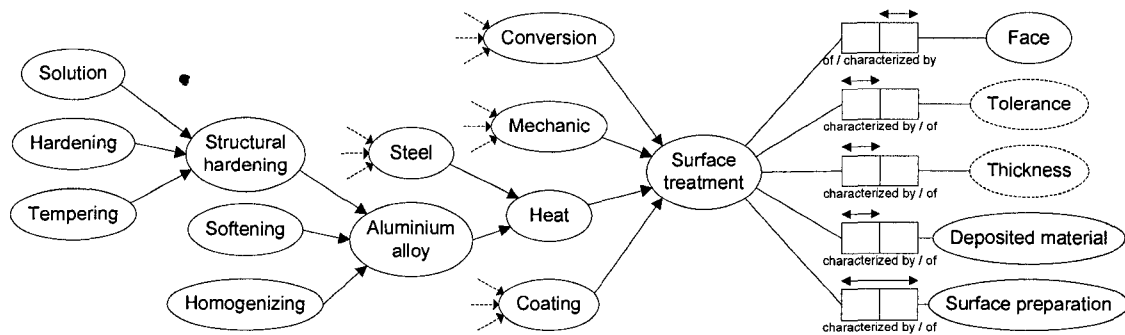


Figure 3.10 Partial schematical representation of the surface treatment feature

3.3.2.5 Surface quality

The surface quality of working and nonworking surfaces needs to be defined. The ISO standard uses many parameters to define the surface quality (Figure 3.11): the main one being the roughness value. Other parameters can be added to the feature to specify the process, like: basic length, manufacturing process, lay direction, waviness, etc.

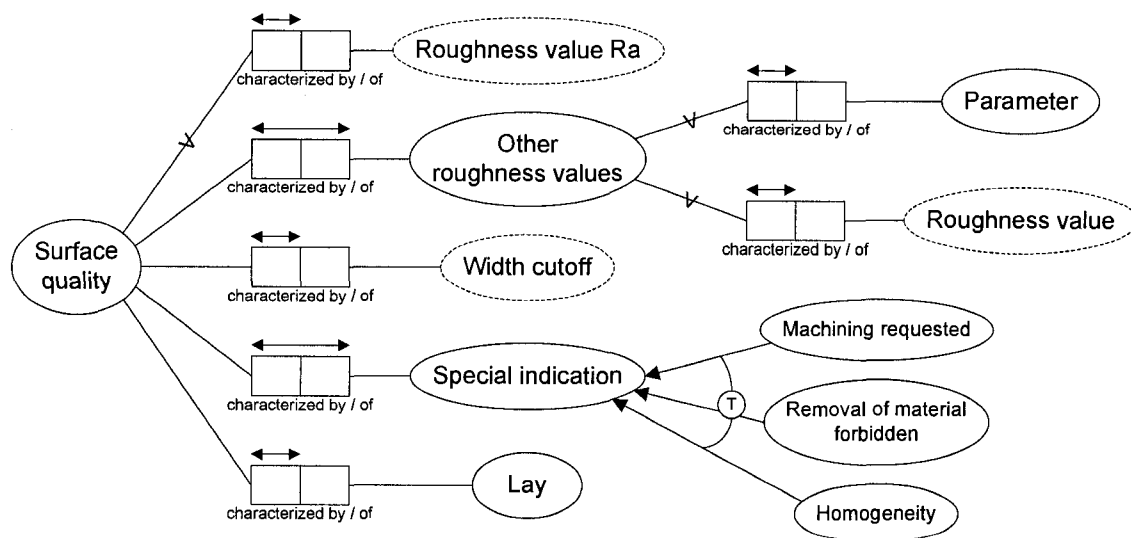


Figure 3.11 Schematical representation of the surface quality feature

3.4 Software modeling

3.4.1 Modeling of a component in the data structure

The data structures developed previously has been used to create a software based on the engineering platform OpenCASCADE (OpenCASCADE 2007). OpenCASCADE is an open source software development platform for 3D computer aided engineering. This architecture of this data structure allows to follow the product during all its development process.

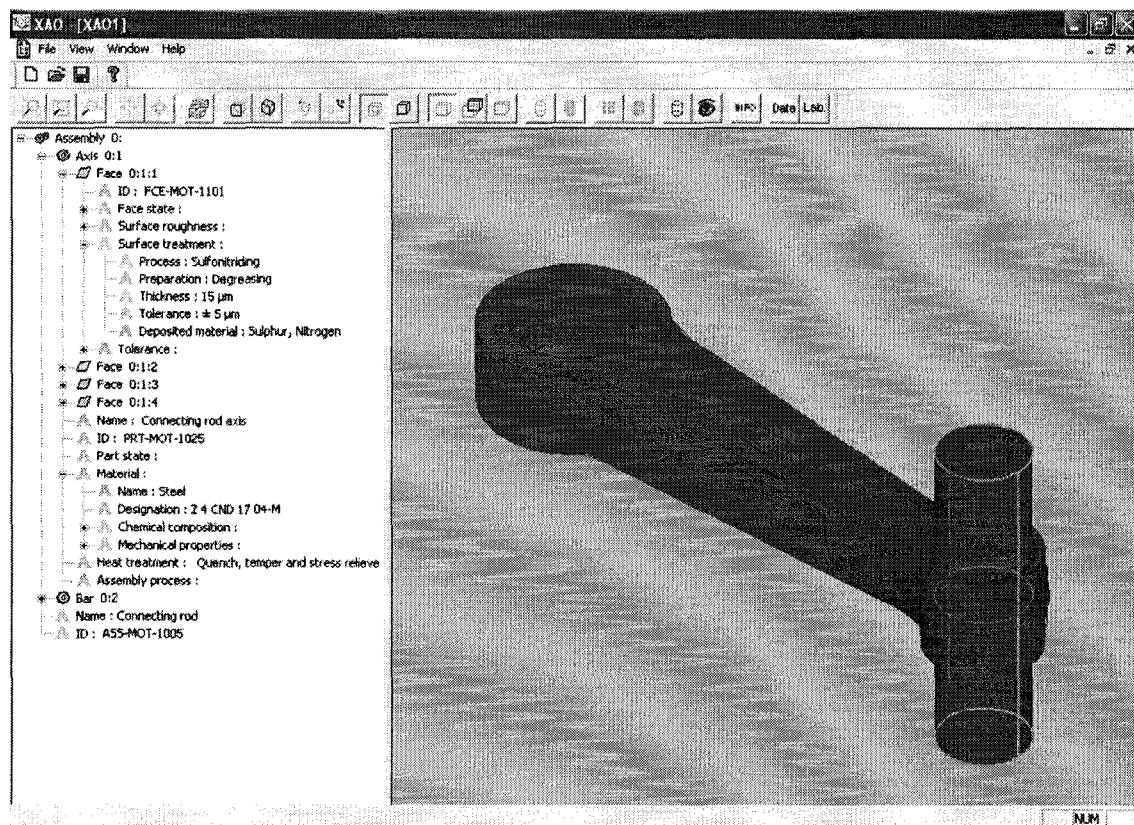


Figure 3.12 Modeling of the data structure in the software

Software modeling requires the creation of various classes, each one is responsible for a part of information on the topological structure. Information can be stored in any level of the data structure (Figure 3.12).

The software analyzes the component, decomposes it into different sub-components, then it extracts the faces from each part. This decomposition in entities allows to build the topological structure in the tree structure of the left view.

The screenshot shows a 'Component information' dialog box with the following fields and options:

- Part:** A text field containing 'Axis'.
- Face ID:** A text field containing '0:1:1'.
- Save:** A button in the top right corner.
- Tabs:** A row of tabs labeled 'Characteristics', 'State Graph', 'Surface roughness', 'Surface treatment', and 'Tolerance'. The 'Characteristics' tab is currently selected.
- Process:** A section containing two dropdown menus:
 - Surface treatment:** Set to 'Sulfonitriding'.
 - Surface preparation:** Set to 'Degreasing'.
- Characteristics:** A section containing:
 - Deepness:** A text field with '15' and a unit dropdown set to 'µm'.
 - Tolerance:** A text field with '5' and a unit dropdown set to 'µm'.
 - Deposited Material:** Two dropdown menus, the first set to 'Sulphur' and the second set to 'Nitrogen'.

Figure 3.13 Features view used for data entry

Using a view dedicated to data entry (Figure 3.13), the user can easily set information about the entities. This view is managed by features which allow to differentiate relevant information about the selected entity according to the specific characteristics of each stakeholder field. These data are gathered in various tabs in order to facilitate the used of the software. The various choices at the disposal of the user are preset (but also allowing edit) according to the needs and habits of the company, to accelerate information entry. These data are also directly managed by the features and the rules mentioned in the previous part: in the case of Figure 3.12, the choice of the surface treatment process (sulfonitriding) involves automatic adjustments.

3.4.2 Software purpose

In the proposed integration model, the designer is able to develop the product and also to enter data concerning the development process. After part design is finished, the user can set information concerning surface roughness via the data view (Figure 3.14). These data should be pre-set using a standard feature according to manufacturing knowledge.

A feature will automatically verify the compatibility of part attributes (selected process, tolerance, etc) with the wished roughness.

Component information

Part: Axis Face ID: 0:1:1 Save

Characteristics State Graph Surface roughness Surface treatment Tolerance

Criterion

Roughness value Ra: 0.8 μm

Other roughness values: Rz 0.4 μm

Width cutoff: 15.00 mm

Special Indication: ☒ Machining requested

Lay: ☐ Perpendicular striates

Striate direction

Figure 3.14 Surface roughness features view

The operation is similar for the surface treatment feature shown Figure 3.13, and the pre-setting of data is also provided by a standard feature to facilitate the work of methods engineers.

Like the designer needs to think about tolerance between attributes during first stages of the development, he can also directly enter the relative data. Figure 3.15 shows the features used for the tolerance entry. The feature allows the selection of the datum in the topological structure using the main view of the part.

The screenshot shows a 'Component Information' dialog box with the following fields and options:

- Part:** Axis
- Face ID:** 0:1:1
- Save** button
- Tolerance** tab is selected, with other tabs: Characteristics, State Graph, Surface roughness, Surface treatment.
- Characteristics:**
 - Tolerance:** Concentricity (indicated by a concentric circle icon)
 - Tolerance value:** 0,03 mm
 - Modifiers:** Maximum material condition (indicated by an 'M' in a circle icon)
- Datum** section:

Dat. 1:	A	0:1:4	Mod.	(M)
Dat. 2:			Mod.	
Dat. 3:			Mod.	
Dat. 4:			Mod.	

Figure 3.15 Tolerance features view

This information can be easily re-used by method engineers and by manufacturing stakeholders and it eases the generation of the various data sheets. In fact, data are digitally linked to the topological structure, that is not integrated in actual software.

3.4.3 Prospects

Such access to the product data during the development process allows to automate numerous activities: manufacturing data sheet generation, assembly and disassembly data sheet generation, draft generation, etc. The main relevance is that these value-added data are the base of many analyses (manufacturability, assemblability, quality, impact on cost, processes capability, serviceability, recyclability, etc). One can easily use them to assist designers about aided decision or management of incompatibilities. The stakeholders will be able to know the impacts of there choices and to make the best ones early in the development process.

3.5 Conclusion

In this paper, we build a new data base representation model using features to represent the development lifecycle. Such information management allows a better data integration within the digital mock-up: saving of time during the data entry, information

is accessible throughout the product development cycle and filtering according to the stakeholder's field, data are linked directly to their corresponding entity in the data base, etc. Current CAD and CAM systems are not able to link value-added information to the geometric model, but these data are required for the design process to take into consideration impacts of design choices. Our model has been implemented with the object-oriented programming C++, using OpenCASCADE libraries (display, topological structure development, data base management), to validate the methodology. Stakeholders are able, during the development process, to model the part geometry and to add corresponding attributes. Our model also allows to inform about the state of each entity during all the lifecycle (manufacture, assembly, dismantling, maintenance, recycling, etc). The prospects for this work are thus numerous because it can be used with all types of data. The system could assist the generation of an assembly or manufacturing data sheet. It appears also very useful in the field of decision aided design: an interface based on features and data of resources (machines, characteristic of the processes, etc) would allow to indicate to the user the relevance of his choices and their repercussion on the product development cycle.

3.6 References

- Anwer N, Mathieu L, Elmaraghy W (2005) Apport des ontologies de spécification et de vérification géométrique des produits pour la conception intégrée. Congrès de conception et production intégrées, Casablanca, Maroc.
- Chang T-C, Wang H-P, Wysk RA (2006) Computer-aided manufacturing, 3rd edn. Pearson Prentice Hall, Upper Saddle River, NJ.
- Chep A, Tricarico L (1999) Object-oriented analysis and design of a manufacturing feature representation. International Journal of Production Research 37(10):2349-2376.

- Coma O, Mascle C, Veron P (2003) Geometric and form feature recognition tools applied to a design for assembly methodology. *Computer Aided Design* 35(13):1193-1210.
- Deneux D (2002) Méthodes et modèles pour la conception concourante. Laboratoire d'Automatique de Mécanique et d'Informatique industrielles et Humaines - Université de Valenciennes et du Hainaut Cambrésis, HDR thesis.
- Dimov SS, Brousseau EB, Setchi R (2007) A hybrid method for feature recognition in computer-aided design models. *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* 221(1):79-96.
- Dong J, Vijayan S (1997) Features extraction with the consideration of manufacturing processes. *International Journal of Production Research* 35(8):2135-2155.
- Gao S, Shah JJ (1998) Automatic recognition of interacting machining features based on minimal condition subgraph. *Computer Aided Design* 30(9):727-739.
- Gardan Y (2003) CAO : vers la modélisation fonctionnelle. *Techniques de l'Ingénieur*, 20.
- Grayer AR (1976) A computer Link Between Design and Manufacturing. - University of Cambridge, PhD.
- Halpin T (1995) Conceptual schema and relational database design, 2nd edn. Prentice Hall Australia, Sydney.
- Jabbour T, Mascle C, Maranzana R (1998) A database for the representation of assembly features in mechanical products. *International Journal of Computational Geometry & Applications* 8(5-6):483-507.

- Lee K, Gossard DC (1985) Hierarchical data structure for representing assemblies: part 1. *Computer Aided Design* 17(1):15-19.
- Miao HK, Sridharan N, Shah JJ (2002) CAD-CAM integration using machining features. *International Journal of Computer Integrated Manufacturing* 15(4):296-318.
- Noort A, Hoek GFM, Bronsvoort WF (2002) Integrating part and assembly modelling. *Computer Aided Design* 34(12):899-912.
- OpenCASCADE S.A.S. (2007) *Open CASCADE Technology*, www.opencascade.org.
- Paoluzzi A, Pascucci V (2003) *Geometric programming for computer-aided design*. John Wiley & sons, Hoboken.
- Quatremer R (1996) *Construction mécanique : précis*, 5e edn. AFNOR, Nathan, Paris.
- Regli WC, Gupta SK, Nau DS (1995) *Extracting Alternative Machining Features: An Algorithmic Approach*. Research in Engineering Design, Springer-Verlag London Limited 7:173-192.
- Shah JJ, Rogers MT (1988) Expert form feature modelling shell. *Computer Aided Design* 20(9):515-524.
- Teissandier D, Couetard Y, Gerard A (1999) Computer aided tolerancing model: Proportioned assembly clearance volume. *Computer Aided Design* 31(13):805-817.
- Tollenaere M (1998) *Conception de produits mécaniques : méthodes, modèles et outils*. Hermes, Paris.

CHAPITRE 4 : REPRÉSENTATION DES DONNÉES « MÉTIER »

Dans le chapitre 3, nous avons introduit la structure de données permettant l'intégration de l'information « métier » à la structure topologique du produit. La structure a notamment été représentée schématiquement à l'aide du formalisme NIAM.

Ce chapitre complète brièvement le chapitre précédent en exposant la technique d'implémentation des features qui ont été développés afin de permettre leur extraction dans l'application développée au chapitre 5.

4.1 Procédés de développement de produit

Au chapitre précédent ont été présentées des expansions partielles de l'objet *procédé* et de ses sous-objets. Nous allons détailler ici la taxinomie de certains *features* caractérisant les procédés de fabrication.

La Figure 4.1 décrit la taxinomie complète du procédé d'usinage. L'objet *procédé*, qui apparaissait à la Figure 3.3, peut être décomposé en sept principaux sous-objets, étant également composés de sous-objets. Chaque *feature* d'usinage caractérise le procédé de fabrication et peut être relié à(aux) l'entité(s) géométrique(s) et ses paramètres de fabrication.

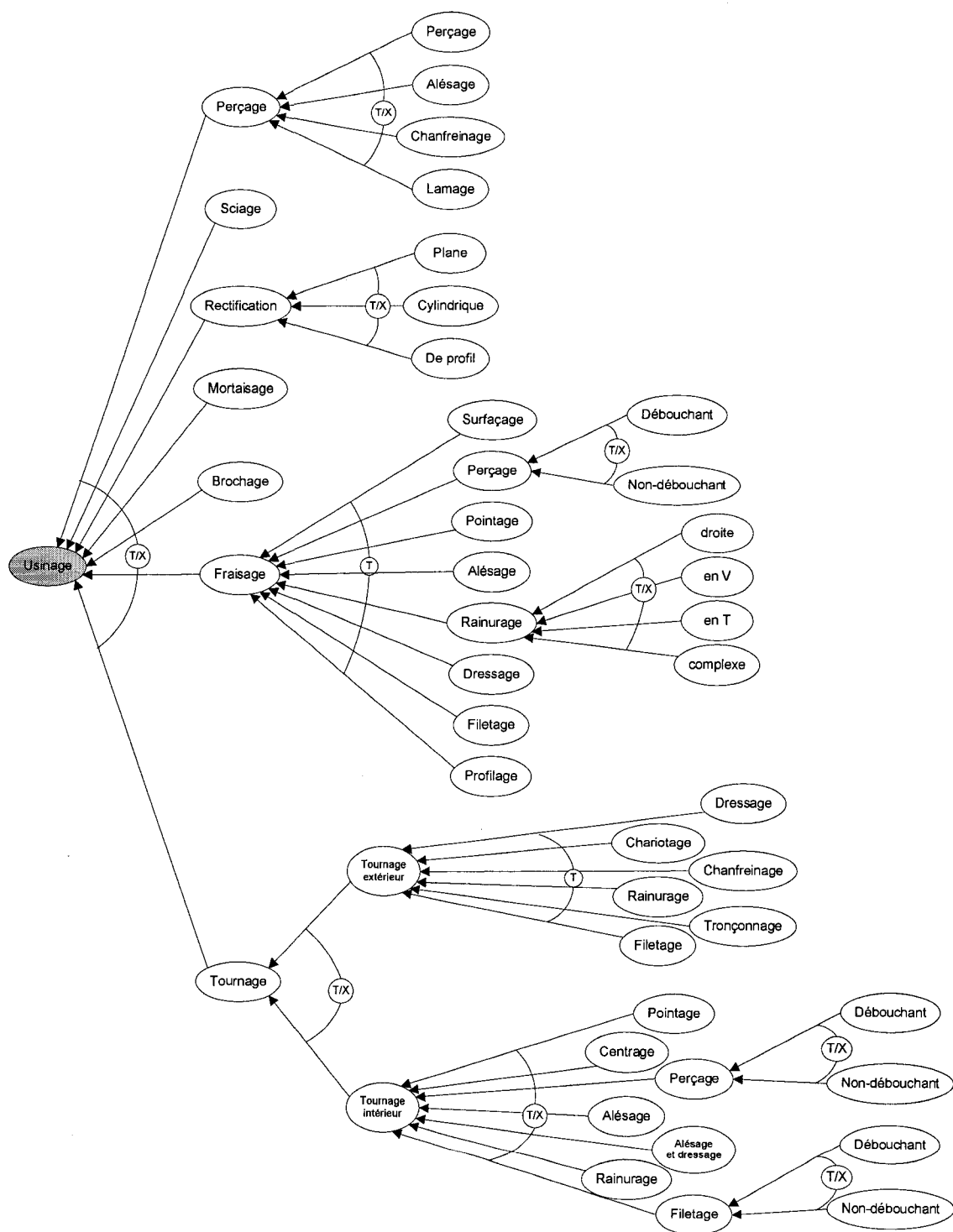


Figure 4.1 Taxinomie du procédé d'usinage

On peut également obtenir la taxinomie du procédé de traitement de surface précédemment présenté à la Figure 3.10 : l'Annexe C, représentant les caractéristiques du *feature* de traitement de surface, expose également les sous-objets de ce procédé.

4.2 Implémentation logicielle des features technologiques

L'implémentation logicielle est basée principalement sur une librairie proposée par la plateforme OpenCASCADE : la librairie OCAF (*Open CASCADE Application Framework*).

Cette librairie permet de spécifier et d'organiser les données utilisées dans le cadre d'un logiciel de CAO. Elle propose une infrastructure capable de lier des données à n'importe quel élément topologique. La Figure 4.2 présente un exemple réalisé à partir de cette librairie. C'est principalement cette propriété que nous avons utilisée dans le développement du premier article.

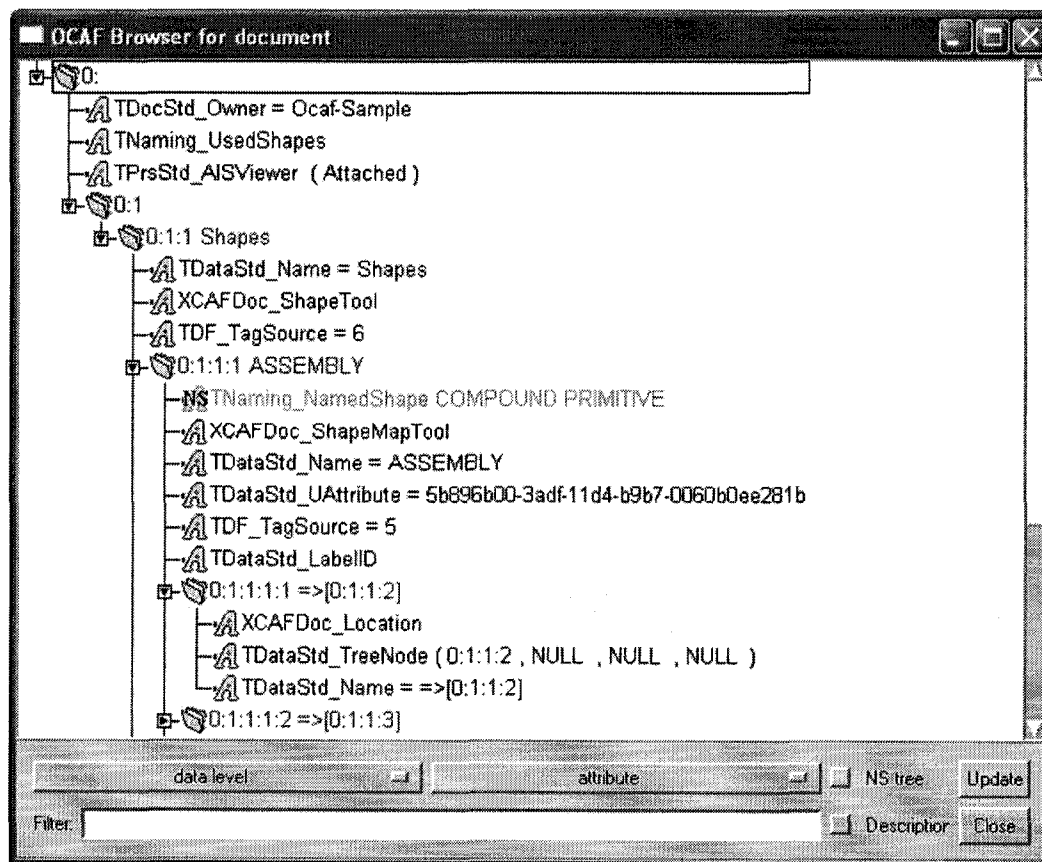


Figure 4.2 Exemple de structure générée grâce à la librairie OCAF

Nous avons pris pour exemple le *feature* d'état de surface (présenté à la Figure 3.11), qui est caractérisé par :

- La rugosité R_a ,
- Les autres valeurs de rugosité (R_y , Z_T , etc),
- La valeur de la longueur de base,
- Les irrégularités de surface,
- Les indications spéciales.

Les attributs sont directement créés dans la base de données (Figure 4.3) sous la forme de variables associées à une unité et liées à l'entité géométrique qu'ils caractérisent. Les types de rugosité sont stockés dans un label spécifique.

Pour traduire les caractéristiques des irrégularités de surface et les indications spéciales, nous avons introduit nos propres objets, offrant une liste de pré-sélection des attributs. Cela permet d'offrir un choix de paramètres à l'utilisateur et d'accélérer la saisie des informations.

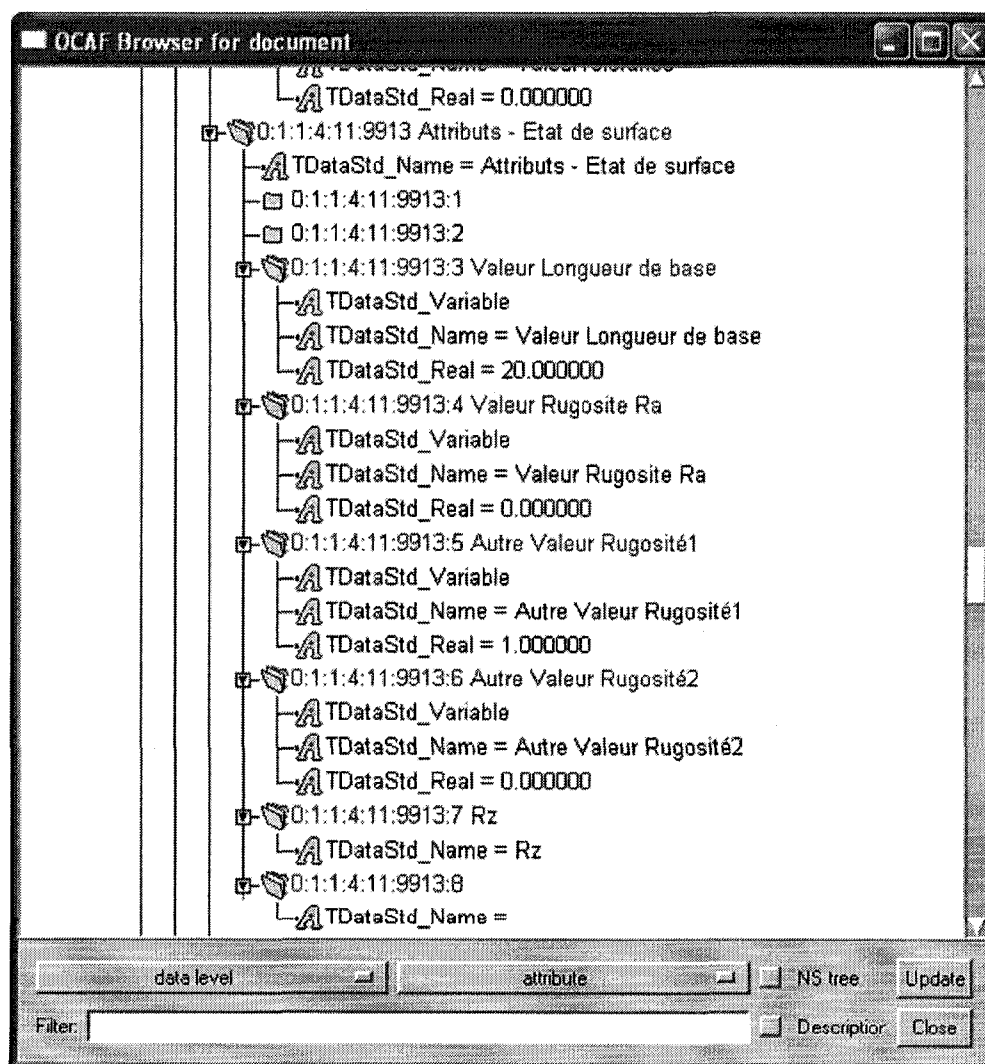


Figure 4.3 Représentation du *feature* d'état de surface dans la base de données

Le *feature* état de surface est caractérisé par l'attribut TDataStd_EtatDeSurface, lui-même composé de différents sous attributs :

- TDataStd_Variable longueurDeBase
- TDataStd_Variable rugosite
- TDataStd_Name typeRugosite1
- TDataStd_Variable rugosite1
- TDataStd_Name monTypeRugosite2
- TDataStd_Variable rugosite2
- TDataStd_IndicationSpecialeEnum, caractérisé par ses attributs :
- TDataStd_USINAGE_EXIGE
- TDataStd_ENLEVEMENT_DE_MATIERE_INTERDIT
- TDataStd_MEME_ETAT_DE_SURFACE_POUR_TOUTE_LA_PIECE
- TDataStd_NULL
- TDataStd_StriesEnum mesStries, caractérisé par ses attributs :
- TDataStd_PARALLELE
- TDataStd_PERPENDICULAIRE
- TDataStd_CROISE
- TDataStd_MULTIDIRECTIONNEL
- TDataStd_CIRCULAIRE
- TDataStd_RADIALE
- TDataStd_PARTICULIERE
- TDataStd_NULL

Dans le cas où la définition du *feature* nécessite une référence à une entité ou un *feature* étranger, on crée un label de liaison dans la structure de données pointant vers sa référence. C'est notamment le cas du *feature* de tolérance qui peut admettre comme paramètre une ou plusieurs références spécifiées (datum).

4.3 Conclusion

L'implémentation des *features* en C++ va permettre, en utilisant le même langage, d'avoir accès aux données du produit. Les algorithmes développés dans le cadre du prochain chapitre vont permettre de traiter cette information métier.

CHAPITRE 5 : ASSISTING DESIGNER USING FEATURE MODELING FOR LIFECYCLE

Soumis à *Computer-Aided Design*

Édition Elsevier

Ce chapitre présente la deuxième partie du travail de recherche. Dans la continuité de l'article précédent, nous étudions ici l'utilisation de l'information technologique relative aux entités du produit afin d'en évaluer l'impact sur les différents paramètres du cycle de vie. L'objectif principal est de déterminer la viabilité des choix du concepteur et de gérer les incompatibilités entre les caractéristiques des processus de production.

Abstract:

Integrated product design imposes to integrate constraints concerning development process as early as possible in the design product process. The main goal is to minimize the changes of design and to involve costs, but also to improve quality and to reduce the delays. To achieve this, information on manufacturing and assembly should be accessible by the actors of the lifecycle. The design for X solution, developed in this study with the OpenCASCADE geometric kernel, allows the designers to get a feedback on the viability of their design choices. The implemented method automatically evaluates both the efficiency factors and the incompatibilities between development characteristics, in order to optimize the design of the product.

Keywords:

Feature Modeling – Concurrent Engineering – Design For X – Product Data Management.

5.1 Introduction

Product development systems have evolved deeply since the 1980s with for main objective, the improvement of the productivity (decrease the costs and the product cycles without reducing the quality of the product). In addition, the manufacturing industry recently took into consideration new concepts like sustainable development. These considerations arise from several factor listed bellow:

- Globalization.
- Diminution of the time to market.
- Computer tools combined to the facility to communicate (and exchange information).
- Sustainable development principally divided among environment and human factors.

Companies require increasingly a better transferability of the information between the different actors of a project to improve the development process and for issues such as certification, legal requirements, customer requirements, warranty and service. That is why production methods are more and more based on Concurrent Engineering (CE) and Integrated Product Development (IPD). CE and IPD allow to take simultaneously into consideration all stages of the product lifecycle. This approach, permitting parallel decisions and development, relies on the sharing of multidisciplinary information.

Each product study needs, before the development process, an estimation of many parameters. These parameters will gradually be adjusted according to calculations or simulations. IPD enables a gradual design of the product and the refinement of every

specification considering influences on the final assembly and the different development tasks. Each decision of the designer impacts on the next stages of the product lifecycle. The designer should therefore be aware of the impacts of his choices. One must also take into account that the impact of changes on the costs is exponential throughout the development cycle. The costs of changes and the constraints due to specifications of the lifecycle (employed process, usability, etc.) force to make the best choices early in the development process. Furthermore, it is necessary to increase the designer's awareness of his decisions' impacts on parameters like time to market, cost, quality, reliability, maintainability, recyclability and human factors. As a consequence, stakeholders need a tool, which would inform them on their choices and their effects on the various parameters of the product development.

This concept has been introduced by Boothroyd and Dewhurst [1], followed by many other authors [2-4], and it is mainly based on the product design for manufacturing and assembly (DFM and DFA). Recently, studies have been focused on many other constraints (design for environment, design for reliability, design for disassembly, design for maintainability, design for lead-time, design for human factors, etc.), merge into the term "design for X" (DFX).

In this context, CAD model becomes the foundation of our project, allowing interaction between each stakeholder. Unfortunately, current softwares and hardware solutions do not allow the integration of such constraints and non-geometrical data so early in the development process. Most of the present optimising product solutions are obsolete. They are based on excellent manual methods but the possibilities with a digital mock-up and CE are very restrictive.

Our approach is able to avoid these restrictions by using an integrated data structure earlier in the development process. The scope of this paper is thus to propose a CAD environment that assists the designer in providing feedback on the product, taking into account the development process. First, we present the data structure which enables the

integration of product information and grants access to the data to any stakeholder during all the lifecycle. Then, we analyse information flows between these actors before to explain two results of integrated applications (decision aided and incompatibility management), using feature modeling.

5.2 Related works

Analysis of current manufacturing methods shows that information is the base of the product development. Computer-Integrated Manufacturing (CIM) concept describes a new approach, including advanced manufacturing technologies from design to management of production. It is a way for every actors of the enterprise to communicate with others. The concept is based on a unique computer database [5] gathering all product information, as shown in Figure 5.1. It allows thus a good integration of enterprise resources and improves communications between departments.

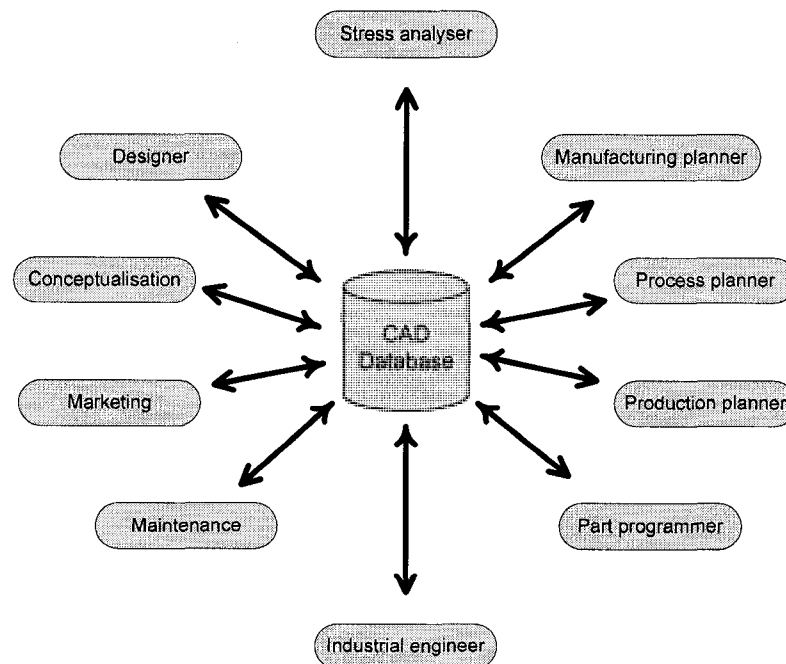


Figure 5.1 Integration of total engineering knowledge

The design methods have been developed with the evolution of modeling methods (CSG modeling, Brep modeling and more recently Feature Based Modeling), but all these methods are linked by the necessity of store more and more data in the topological framework. The hole feature, used by all current CAD/CAM softwares, is a good example of this needing information: designing a hole with its specific feature allows to characterize manufacturing process and its parameters, and also to automatically infer other non-geometrical parameters (pre-drilling required, type of machine-tool, etc). In this study, product information can refer to each stage of the development cycle; previous studies dealt with integrated data concerning : product tolerance [6-8], assembly [9-11], machining [12], molding [13], etc.

5.2.1 Feature modelling

Most actual CAD solutions allow to model parts from advance primitive creation functions, called features. Many authors have suggested a definition of a feature [4, 14-16]. It is described in this study as a way to transfer geometric, technological or functional information of an entity between various actors throughout the development process and the lifecycle. Each feature can be specialized according to its own domains (design, manufacturing, assembly, recycling, etc), and it is defined by its dimensions, its topological entities, its specific parameters, and its relationship to other features. Sy [17] proposed a mathematical representation of the features according to the previous definition :

$$F_p = \{T, G, A_p, R\}$$

Where F_p is the product feature, characterized by :

- The pointer T , which links it to the topological structure of the part.
- The geometric information G .
- The attributes A_p of the feature.
- The relationship R to other features attributes geometrical information.

Here are the data implemented in the software to build any feature describing a technological characteristic of the product.

The Feature-Based Modeling (FBM) allows the user to model solids by using such oriented features where many data are pre-set [18]. It became the main geometric modeling systems, because products are not only defined as geometric data, but also by technological characteristics. The FBM method provides a bridge between design step and the rest of the lifecycle, and that is this specificity of the feature we exploit in this paper to support the information transmission.

5.2.2 Aided decision

Decisions take by designer during the design process represent about 70 to 80% of the product manufacturing cost. It is thus essential for designers to get a feedback about impacts of each solution. According to Mandorli [19], this information is characterized by qualitative and/or quantitative evaluation, but it is unfortunately not directly controlled by designers in actual CAD software.

The main application for decision aided has been first developed by Boothroyd-Dewhurst and Hitachi [20] to improve product assembly earlier in the design process. The Design for Assembly (DFA) approach allows to highlight parts of the product where assembly costs are not optimized, calculating a design efficiency coefficient. One can accesses to the analyse evaluating weighting factors concerning various parameters (dimensions, symmetries, assembly process, tools, ease of handling, ease of insertion, etc). Afterwards, Boothroyd and Dewhurst generalized their works in manufacturing process (machining, injection molding, sheet metalworking, die casting, powder metal, etc) with the Design for Manufacture and Assembly (DFMA) [1]. This approach is very interesting because it can be apply to many domains, the principal obstacle is to define weighting factors. However it involves a detailed definition of the product and the assembly sequence, that is not compatible with an interactive design (study is limited to an iterative analyse of changes after the design). The DFMA has then been implemented

in a computer based tool, which allows to calculate theoretical minimum number of parts, assembly time, product cost, etc. But the software solution generates a loss of time and information, because each parameter (technological and geometrical) should be manually input. In addition, inputted data are not linked to the main database, so each change in the design has to be reflected on the DFMA module. The method is thus not adapted to Concurrent Engineering solutions: data are redundant, information transfer is not automatic, and change management is not supported. Many authors have studied the possibility to automate information transfer between the topological structure and the DFMA tool, using feature recognition technology [21, 22] for setting some data (semi-automatic method). A more complete method has been developed by Coma [23, 24], using a form feature recognition tool to extract geometrical data and a fuzzy logic model to extract other required parameters (handling, insertion, etc).

In order to avoid problems relative to information transfers, we propose in this paper a data integration model where information is directly accessible for processing from the design database, according to the Figure 5.1. The main relevance is that data can be easily used by every actors of the development process, which is not possible with classical CAD data structures.

DFMA tools are very popular in engineering design, and many authors have also developed software for assist designers in his work with always the same method: predict various lifecycle parameters integrating data to the part structure, in order to adapt the part design to production facilities. Brissaud [12] built a CAD environment to provide information during the design about the possible locating and clamping regions according to dimensions and tolerances of the part. The module was also able to deduce accessible machining features, intensity of clamping and part stability.

Similar methods have been used to valuate generating costs and producing value during the product development. Deneux [25] define two cost evaluation techniques: parametric methods (based on a statistical correlation between product costs and pertinent

parameters like weight, volume, surface, etc) and analogical methods (based on many cases where costs are known). The parametric method is not appropriate for current aided design because of complexity of product processes. Haasis [26] studied the analytical cost calculation. This methodology requires geometric parameters (which can be extract in our case directly from the topological structure), and the determination of machined and unmachined faces resources considering machining resources. This technique is however limited by unlisted cases in the manufacturing feature and consequences of a feature on the adjacent entities. The author proposed to base the cost evaluation on a knowledge database merging previous cases, but decreasing its adaptability. The model used by Deneux [25] allows to increase the reliability of evaluated factors, even if the part is complex. He based the method on processes knowledge (practice, rules, standards, etc) to estimate cost and value parameters using a loop for each actor concerned.

Mandorli [19], exposes the various feature based technologies and their implication on the user. The designer work varies depending on the degree of integration: from the defined feature where the parameters have just to be interactively set, to feature where properties and control algorithms have to be implemented.

We chose in this paper to use pre-defined feature, in order to facilitate stakeholder's intervention and also to limit annoyances on designer's tasks.

5.3 Information flows

As shown in the Figure 5.1, the Computer-Integrated Manufacturing concept involves various process segments during the development cycle, and the information flow relating to the product concerns more actors throughout the lifecycle. The Figure 5.2 shows various information relating to the product: geometry, resources and also technological data. All these data are used during the development process to build technical drawings or production data sheets, but also to evaluate many parameters.

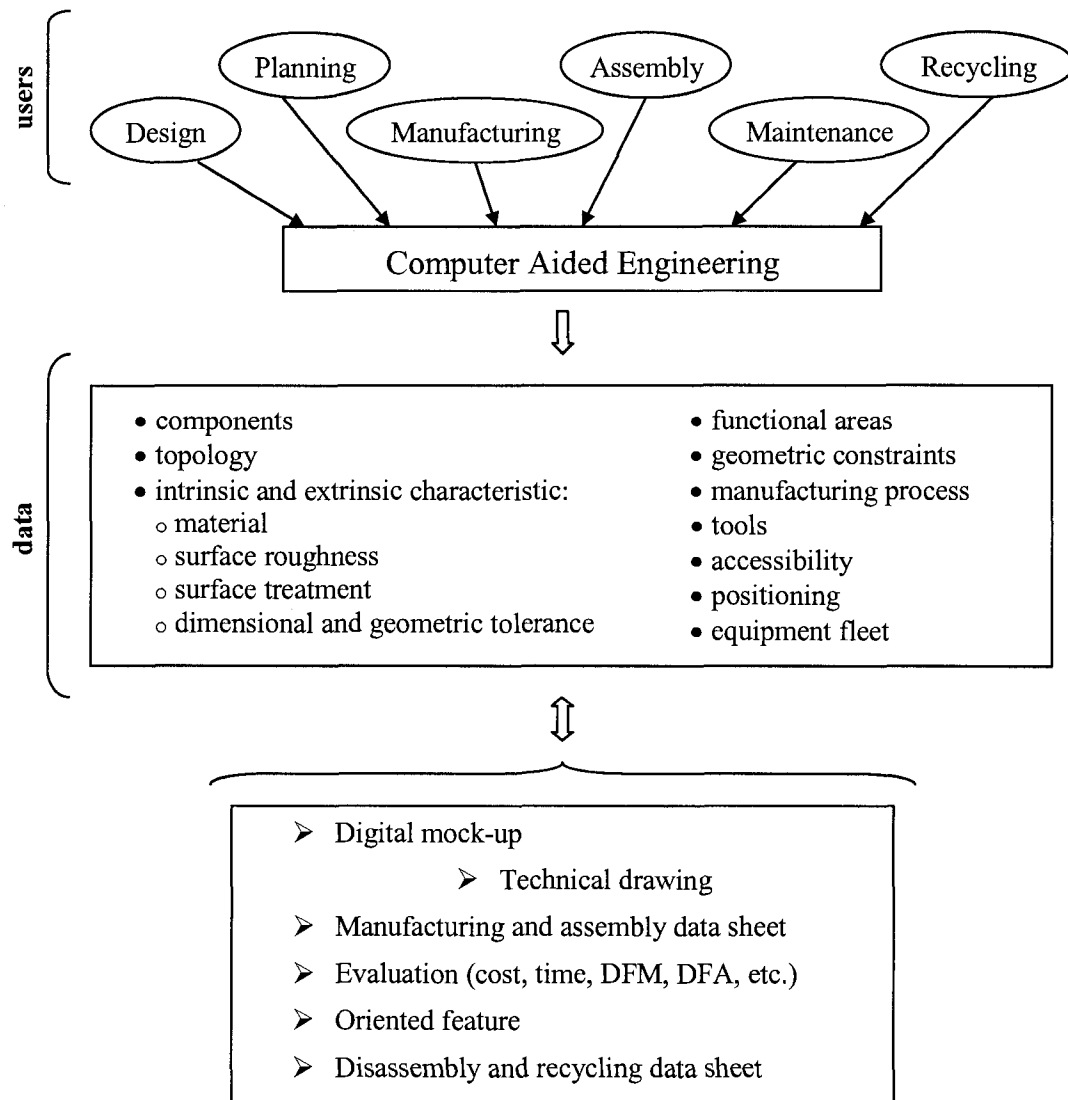


Figure 5.2 Data integration in the development process

Analyse of data flows shows that every actor focus on specific information: the design produces essentially geometric data which are used by the other, the production engineer develops manufacturing data sheet using geometric data and characteristics of the product according to machine resources. Each stakeholder is only concerned by a set of properties (as shown on Figure 5.3) from different entities, which is called the feature

view [27]. The main difficulty is to correctly manage the information transfer to ensure that every actor get upstream data he needs.

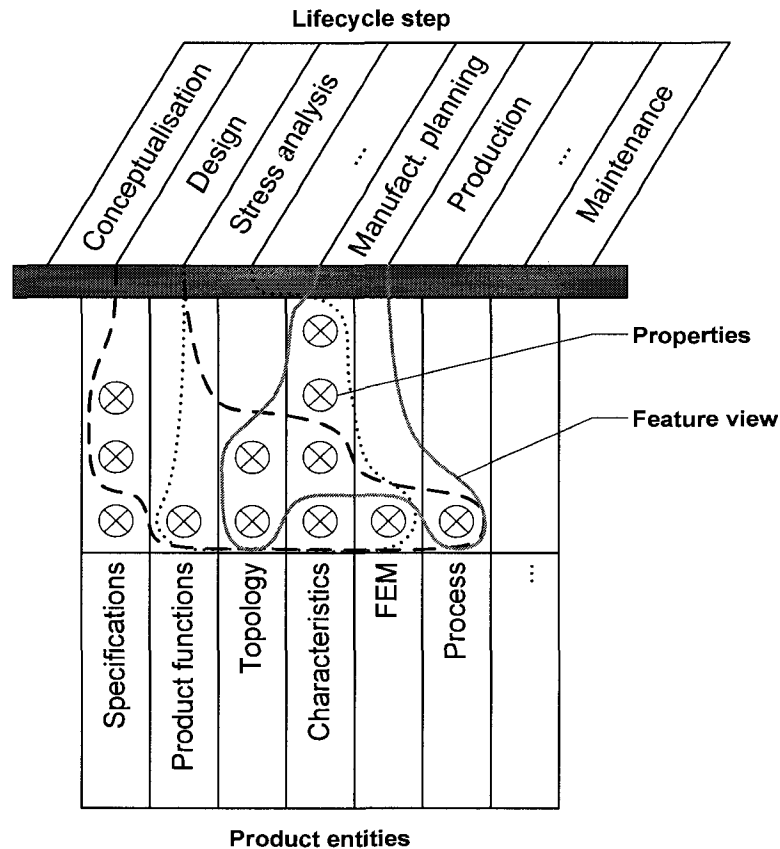


Figure 5.3 Feature views of the product

The study of data exchanges between each stakeholder shows that the conventional design is not adapted to the CIM concept. Information transfer throughout current design methodologies (Figure 5.4.a) is sequential, the designer sets geometrical information in the CAD database and next the stakeholder retrieves the topological framework to add his own information. The main limit is that the sequential transfer forbids backward communication, and reuse of information. As example, the drafting from the digital mock-up for technological data entry (tolerance, surface roughness, etc) is irreversible:

these meta-data are not digitally accessible in current CAD solutions and extraction of information requires manual activity.

The integrated design (Figure 5.4.b) is based on a unique access to databases which allows interaction between data. Contrary to classic CAD database, which don't process technological information during the data entry, the auto-extraction of geometric and technological information enable to automatically deduce various parameters using specific features. This system can thus facilitate the work of actors of the development process.

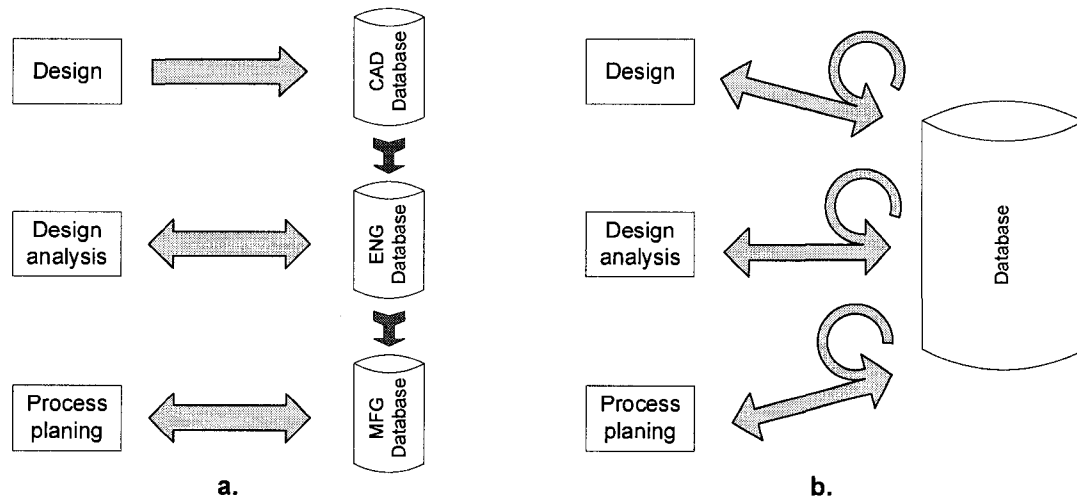


Figure 5.4 Information exchange in conventional (a) and integrated (b) design

The next section will develop the data structure at the base of our study, which provides information exchange between all actors of the lifecycle.

5.4 Data structure

During the lifecycle, data can deal with many domains like: assembly, manufacturing, tolerance, surface treatment, maintenance, recycling. Before developing how model the different processes, we have to introduce how model the data structure of the topology.

We needed a static analysis method able to manage relations and constraints between data (temporal aspects and initial state are not study, so functional and dynamic methods are not required). We used an Object Role Modelling (ORM) method to build the data structure: the NIAM Formalism (Nijssen Information Analysis Method), which allows us to explicitly symbolise the various entities of the product, and also their relations and constraints.

We implemented this data structure in a CAD software using the OpenCASCADE development platform and the C++ programming language. The OpenCASCADE environment allowed us to manage the geometric information and the technological information.

5.4.1 Part and product data structure

The part topology is built from basic elements: vertices (0D element), edges (1D element), faces (2D element) and solids (3D element). These elements allow the decomposition of a part or a product (as an assembly of many parts), which can be represented with NIAM formalism, as shown in Figure 5.5.

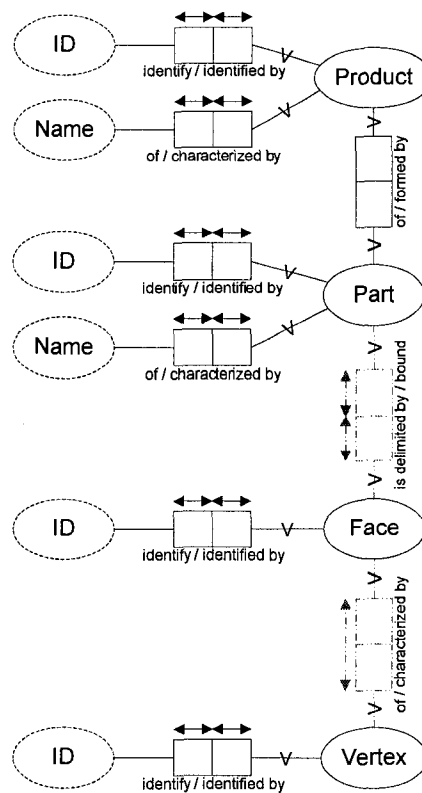


Figure 5.5 Schematized product topology

The study of data flows throughout the lifecycle shows that mostly information is concerning the entity face of the product. Design decisions are actually made according to areas, so most of functional tolerances are relating to faces. In addition, product processes change the state of the part face by face (except for molding and forging, where the generating part model is different). One can also observe the same fact about assembly process where face is the most appropriate entity for the determination of contact, grasping and positioning parts [9]. That is why our data structure is principally based on the entity face, which allows to model the totality of the lifecycle and to include intrinsic and extrinsic characteristics (Figure 5.6).

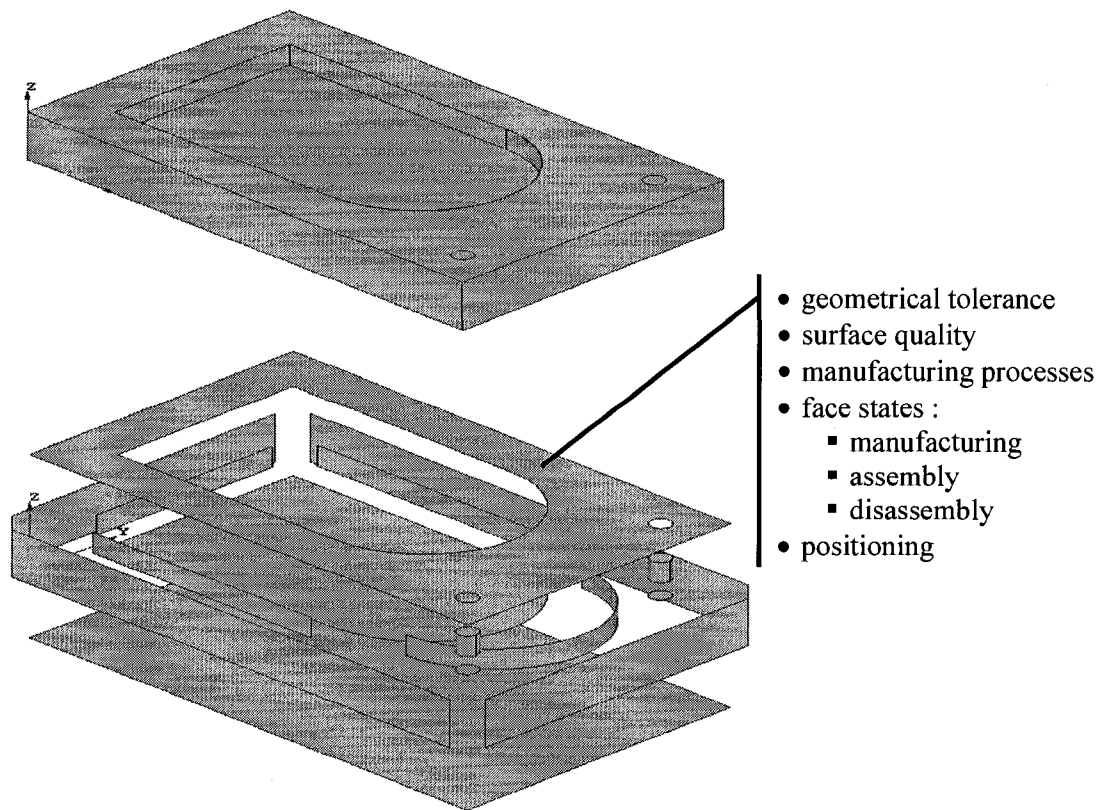


Figure 5.6 Intrinsic and extrinsic characteristics of the face

This data structure representation is particularly adapted for solid modeling systems like Boundary Representation, because of the B-Rep data structure stores basics entities which compose the boundary of a product and how they are interconnected. It is very easy to access to the list of bounding faces from the face entity table [28], and its topological structure is comparable to the adopted structure (Figure 5.5). The Feature-Based Modeling, used in recent CAD/CAM software, is also adapted for such data structure. If topological data are not related to the feature, this modeling function should be linked to a feature recognition tool [23] in order to extract basics entities.

During the modeling stage, the user can handle two kinds of data: geometric data concerning the topological structure, and non-geometric data relating to the lifecycle

process. These data can be set at different stages of the design process and in various ways, these subjects will be developed later. The model is based on the topological resources, which can be generated from the geometric data of the CAD product. The simplified procedures for decomposing the product in sub-products, parts, faces and attributes are written below in pseudocode.

```

Generate_Product_Structure()
Begin
    Initialize the selection of component
    Find the list of active components
    Initialize product iterator
    For each iterated component do
        Decompose_product(product, label)
    Enddo
End

Decompose_Product(product, father_label)
Begin
    If the component has sub-products then
        Decompose_Product(product, label)
    Else
        Add sub-product label to the tree
        Decompose_Part(part, label)
    Endif
End

Decompose_Part(part, father_label)
Begin
    Initialize face iterator
    For each iterated face do
        Add face label to the tree
        Set_Feature(type_feature, label)
    Enddo
    Add part label to the tree
End

```

This algorithm allows extracting the tree structure (Figure 5.7.a and Figure 5.7.b) of a component so as to build the lifecycle process data structure with the non-geometric data.

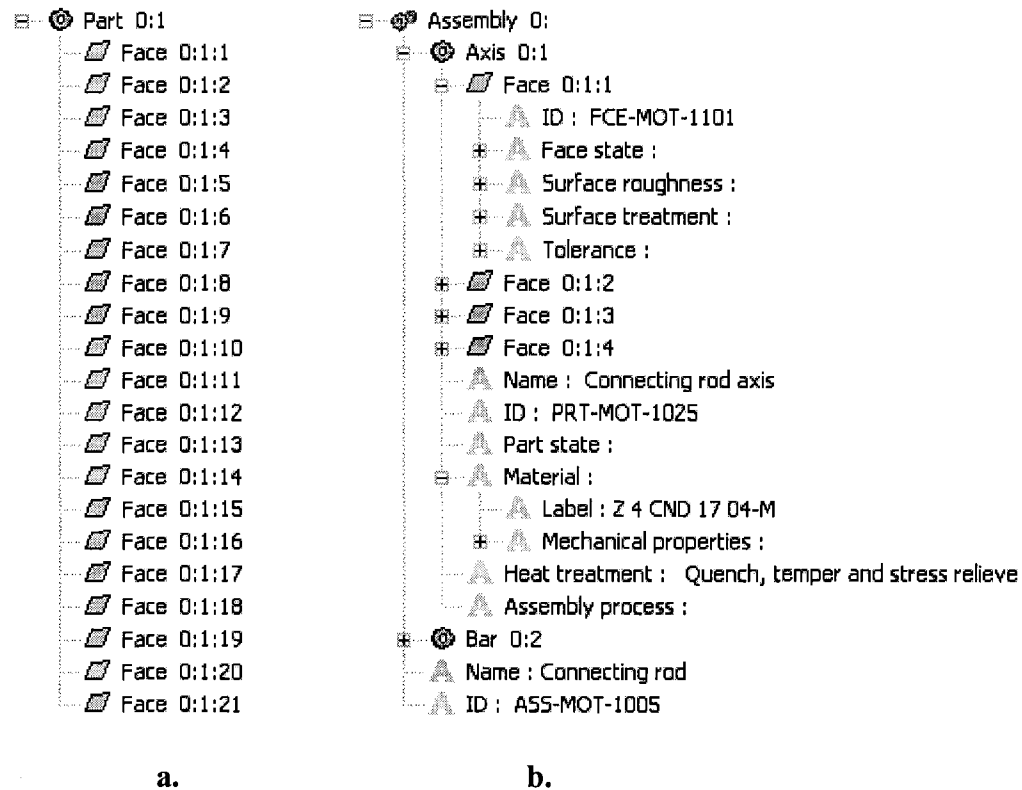


Figure 5.7 Tree structure of a part (a) and an assembly (b)

The tree structure provides to gather technological data (Figure 5.7.b) relating to each entity (face, part, assembly), which will be defined in the next part.

5.4.2 Lifecycle process data structure

The face entity definition in the data structure allows studying the transmission of related information throughout the lifecycle. The different processes used during the product development can be modeled according to the NIAM formalism. Figure 5.8 highlights that manufacturing and assembly processes can be represented globally.

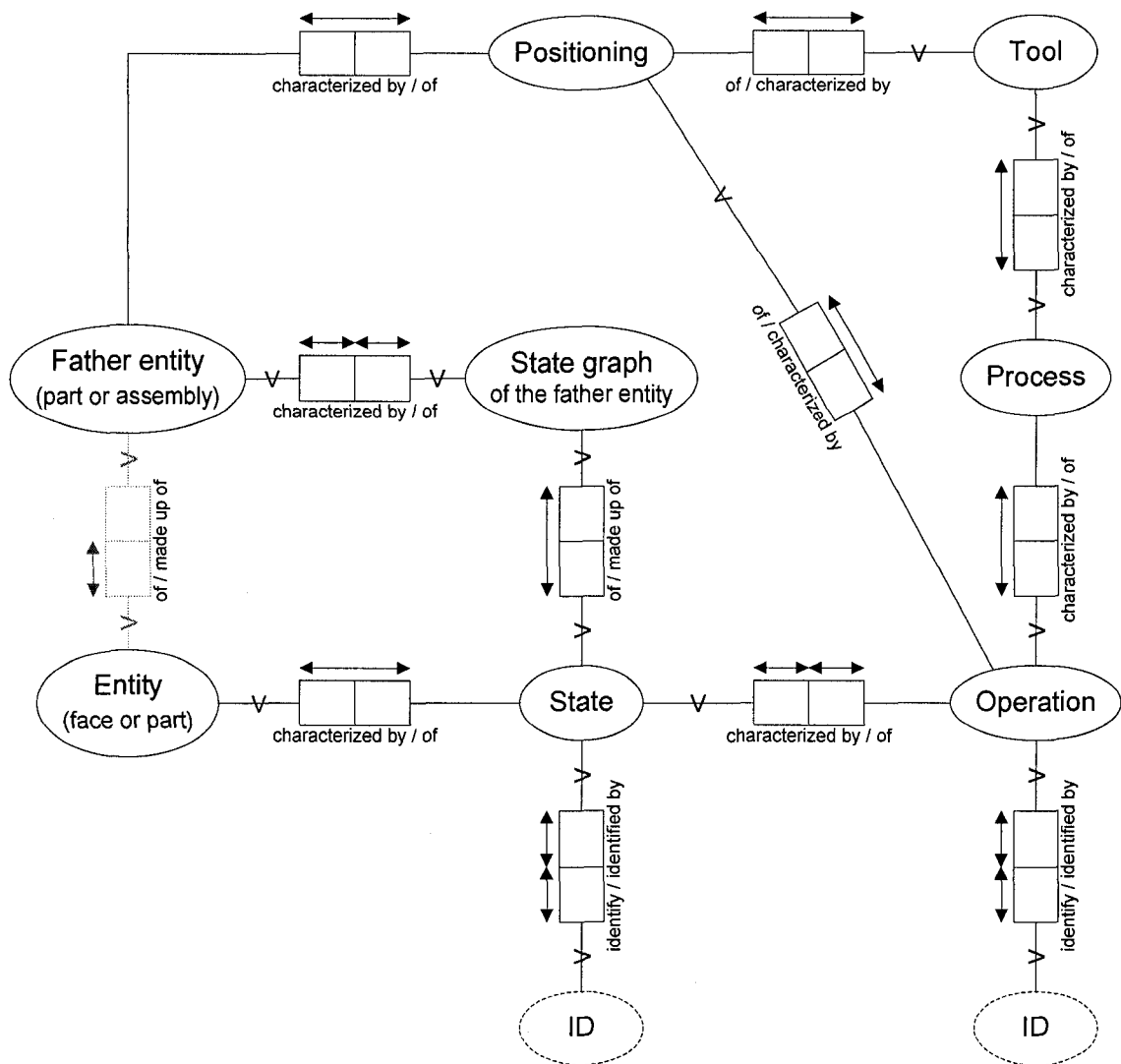


Figure 5.8 Modeling of lifecycle processes

As explain previously, the processes are defined like a sequence of state changes of the different product entities. From this fact, one can define the state of each entity at each step of the lifecycle, from the design to the recycling of the product. That is why on the Figure 5.8, an entity is characterized by its state. States are gathered in the state graph, which represents the state of the father entity in the current process and more generally in the lifecycle process. For the manufacturing process, the state graph corresponds to the route sheet, allowing to detail the different manufacturing steps and which tools and

machines are required for this step. The assembly data sheet is also determined by the state graph of the assembly process, which collects assembly sequences.

Each state of the state graph is associated with an operation, linked to a process. The process object can be divided in many sub-objects according to the case. Figure 5.9 shows detailed manufacturing and assembly processes.

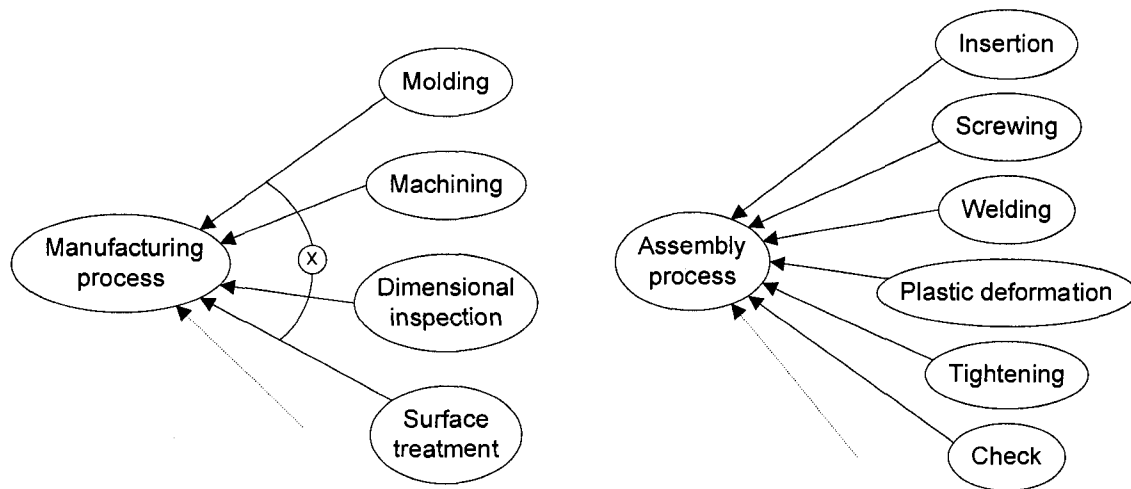


Figure 5.9 Example of subdivision of processes

A process is characterized by its tools and its machine-tools. The link to the father entity is due to the positioning attribute as shown in the Figure 5.8, but the positioning of different entities depends directly on the process operation type. That is why this attribute is linked to the process object. As says previously, this schematization has to be adapted according to the process.

The manufacturing process needs indeed two sub-attributes, linked to the father entity, to define the positioning (Figure 5.10.a): the set-up and the grasping location. The assembly process requires more parameters to be correctly defined (Figure 5.10.b). The same sub-objects are linked to the product entity (positioning and grasping location), but two more attributes are connected to the part entity to characterize its insertion and its grasping location.

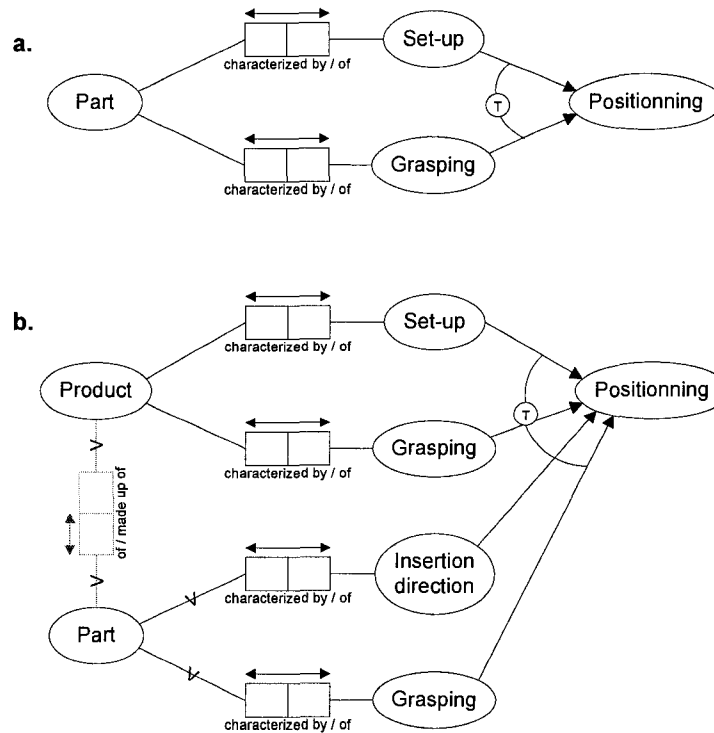


Figure 5.10 Subdivision of positioning attribute according to the process of machining (a) or assembly (b)

5.5 Integrated Engineering

Previous chapters suggest a method, able to allow a good management of information flows. The main aim is to make easier exchange of value-added data between the different actors working on the product. A good integration of information is the starting point of the integrated engineering: data are now available for any processing. This section discusses different methodologies to assist the designers about aided decision or management of incompatibilities.

5.5.1 Decision aid

The responsibility of the designer is to optimize the product design. Engineers take decisions which can affect all the design process, they need thus to foresee consequences

of their choices on the product lifecycle in order to optimise and find the best design. The goal is to provide designers a way to iterate various parameters according to the process shown on Figure 5.11.

There are various ways to inform the stakeholders about impacts of their choices. In this study, an efficiency factor (which can match various parameters) is introduced. It characterizes every used resource. The processing stage evaluates the efficiency factor, extracting useful data from the product data structure. The designer can afterward validate his choice or iterate some part data to optimise the result.

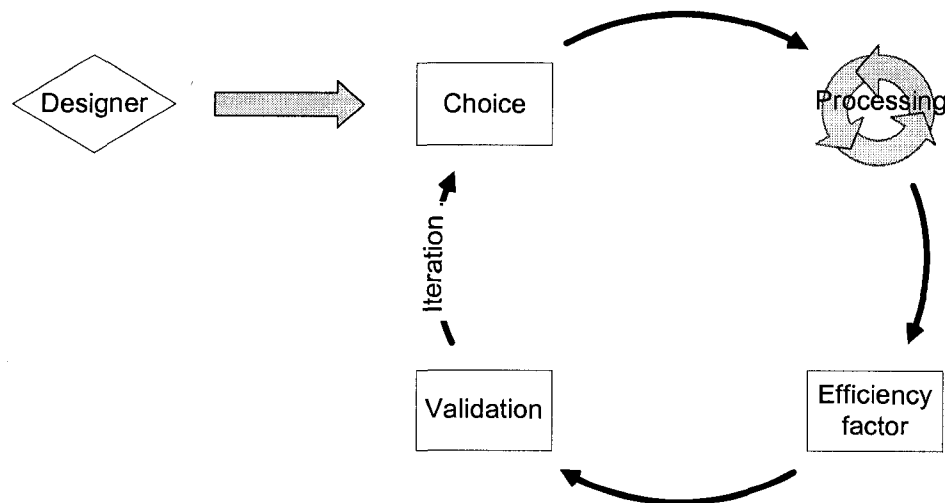


Figure 5.11 Decision aided principle

The efficiency parameter can be related to various domains:

- Cost
- Assemblability (DFA method)
- Manufacturability (DFM method)
- Disassembly
- Maintainability
- Reliability

- Recycling
- Etc

In this study, we will introduce a solution to apply the decision aided principle to the design process.

The previous parameters are linked to intrinsic and extrinsic characteristics of faces or parts (material, tolerance class, modifier, etc) and also to characteristics of processes (manufacturing process, required tools, etc). We estimate the product efficiency parameters from the efficiency of each entity. The final value is obtained by a weighting of these parameters, which can be schematically represented with the NIAM formalism (Figure 5.12).

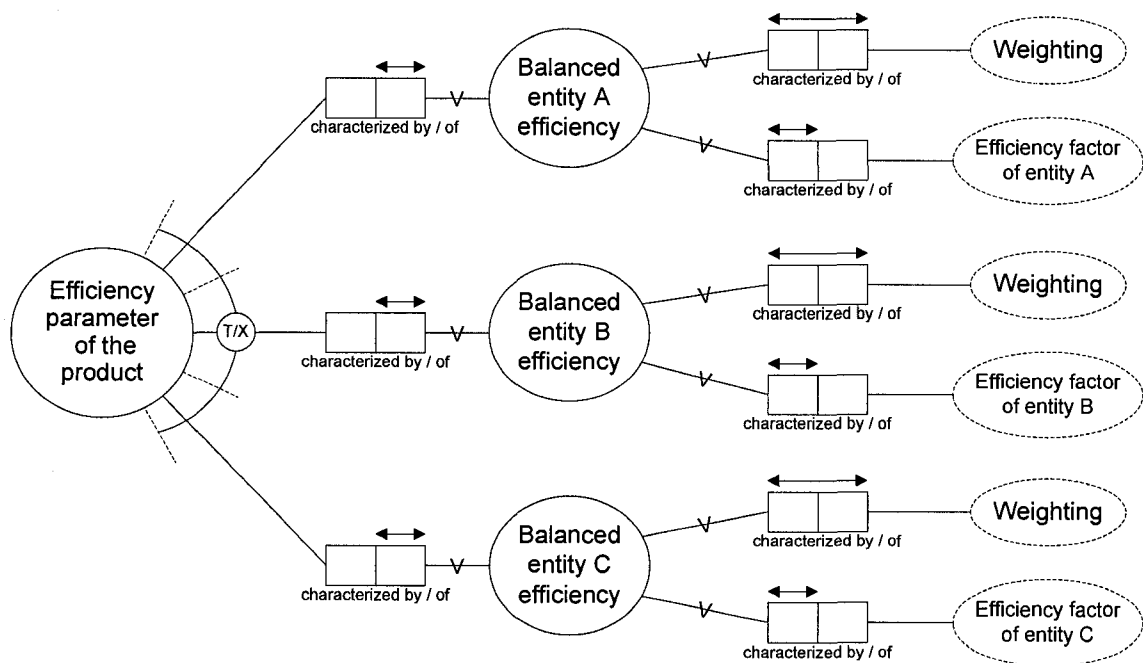


Figure 5.12 Efficiency assessment of the product

The efficiency factor of each entity can be deduced by the same method, according to the extracted information of the data structure.

Entities weighting factors are deduced from various studies of the production process. We need to collect enough information to build a complete table about every parameter. In the literature, we can find such values concerning defined factors and their relationship to an efficiency factor. Boothroyd based his research studies about DFMA on these values and published his results. For example, we can find all information about the selected material according to the process [1]: cost of work material, tensile yield stress, elastic modulus, compressive yield stress, density, etc. Shetty [20] lists some parameters for manufacturing process and their interaction with technical data. The Figure 5.13 shows relationship between the relative cost, the accuracy and the surface finish. This graph allows us to build a table for each value which can be selected by the user. The Figure 5.14 shows how the sub efficiency factors determine the efficiency factor of an entity for the surface treatment and the dimensional tolerance, defined on a previous study. They are extrapolated by a technological feature from the previous tables of values. We gave an example of cost implication but the method can be adapted at every domains previously cited like the recyclability [29]. The only significant point is to build a sufficient table of values.

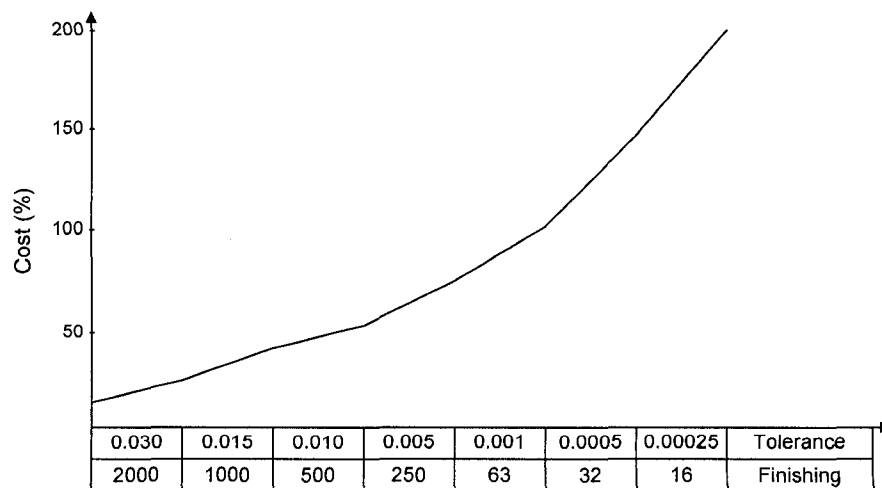


Figure 5.13 General cost relationship of various degrees of accuracy for a number of machining processes, adapted from [30]

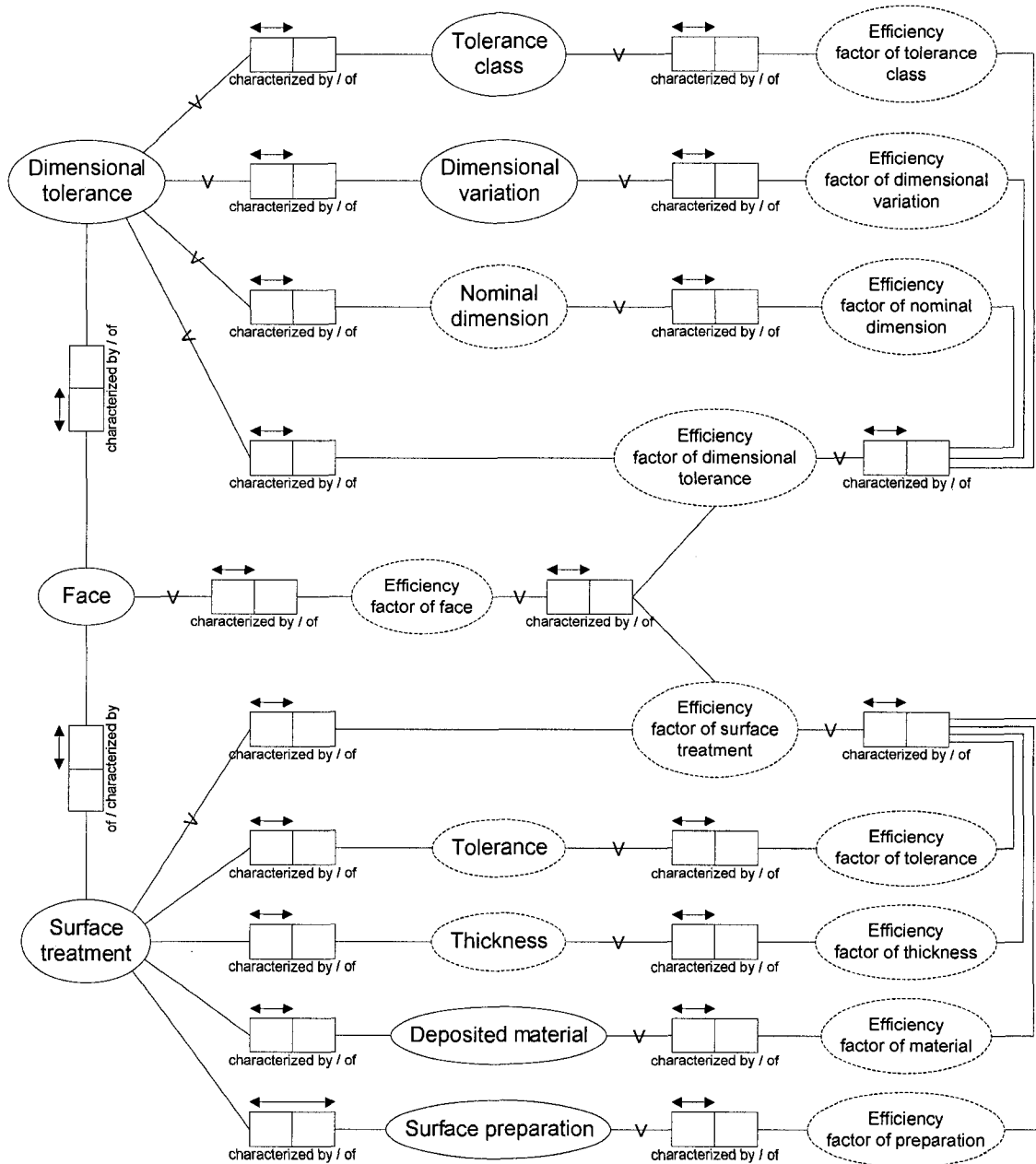


Figure 5.14 Schematically representation of the efficiency factor

This information is essential to the designer to consider the consequence of his choices. However, the goal is not to give him the more exact value of a parameter, but to give him a way to compare various solutions in real-time (sand casting vs die casting,

different roughness values, etc). That is why we standardize all factors from 0 to 10 and we present it to the designer using a user graphical interface (Figure 5.15). He can thus optimise his design according to the factors he wants to improve.

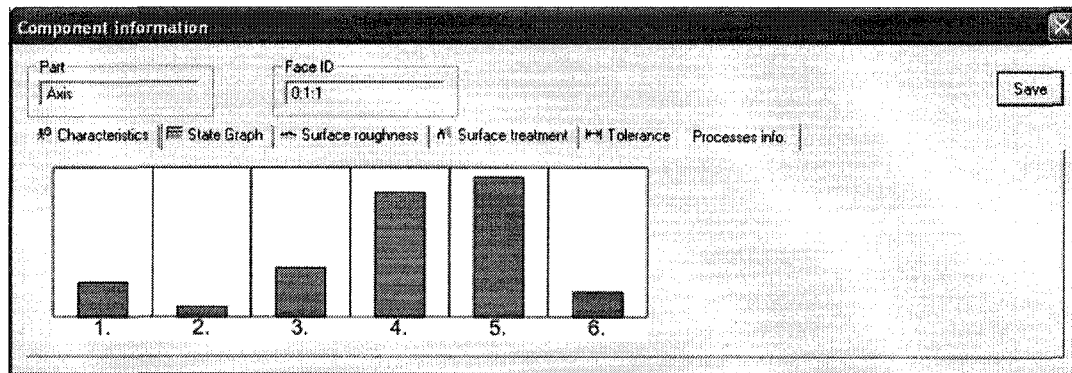


Figure 5.15 Example of user interface for aided decision

5.5.2 Incompatibility management

Integration of information also allows to assist the designer, reporting the incompatibility between technological features. This concept has not the aim to optimise the design, but just to consider every parameters and to highlight better choices. As for the decision aid, we can found in the literature many tables about compatibility between product and process parameters:

- Material vs processes [1].
- Surface roughness vs processes [18, 20].
- Surface treatment vs corrosion.
- Size or complexity vs processes [20]
- Production speed vs processes [20].
- Tolerance vs processes [31].
- Modifiers vs Tolerance [18]
- Number of units vs processes [31].

The large amount of data and parameters interaction requires an incompatibility management tool, which could supervise every rules of the development process.

This study identifies two types of incompatibility: strict and large. The strict incompatibility forbids the use of two parameters simultaneously, whereas the large incompatibility ranks the solutions according to their suitability.

In the first case, we create a rule between each concerned parameter of features. For example, modifiers used to explain tolerance are not compatible with every feature: orientation and position tolerances can be used with maximum material condition (MMC), regardless of feature size (RFS) and least material condition (LMC) but runout and concentricity do not abide MMC or LMC. The rules have to be implemented in the management feature, which extract data entry and disable incompatible parameters (Figure 5.16).

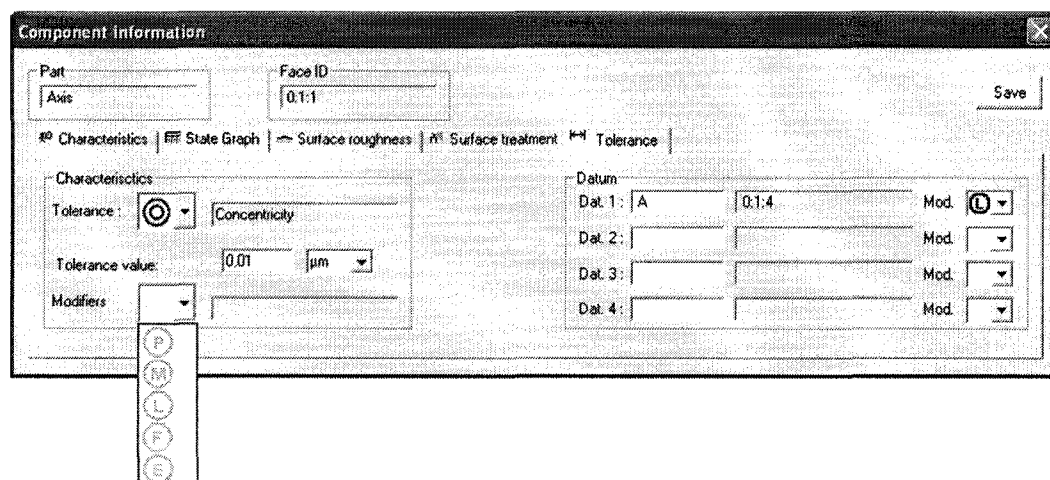


Figure 5.16 User interface using strict management of incompatibility

In the second case, we have to consider more data and more cases, because we regard as the compatibility degree between many characteristics. The Tableau 5.1 shows an example of compatibility degree between two parameters (materials and processes) we used for this study.

Tableau 5.1 Compatibility between processes and materials, adapted from [1]

H = High L = Low X = Incompatibility		Cast Iron	Carbon Steel	Alloy Steel	Stainless Steel	Aluminium and Alloys	Copper and Alloys	Zinc and Alloys	Magnesium and Alloys	Titanium and Alloys	Nickel and Alloys	Refractory Metals	Thermoplastics	Thermosets
Solidification processes	Sand Casting	H	H	H	H	H	H	L	H	L	H	L	X	X
	Investment Casting	L	H	H	H	H	H	L	L	L	H	L	X	X
	Die Casting	X	X	X	X	H	L	H	H	X	X	X	X	X
	Injection Molding	X	X	X	X	X	X	X	X	X	X	X	H	L
	Structural Foam Molding	X	X	X	X	X	X	X	X	X	X	X	H	X
	Rotational Molding	X	X	X	X	X	X	X	X	X	X	X	H	X
Bulk Deformation Processes	Impact Extrusion	X	H	H	L	H	H	L	L	X	X	X	X	X
	Cold Heading	X	H	H	H	H	H	L	L	X	L	X	X	X
	Closed Die Forging	X	H	H	H	H	H	X	H	H	L	L	X	X
	Powder Metal Processing	X	H	H	H	H	H	X	H	L	H	L	X	X
	Hot Extrusion	X	H	L	L	H	H	L	H	L	X	X	X	X
Material Removal Proc.	Machining	H	H	H	H	H	H	H	L	L	L	L	L	L
	Electrochemical Machining	H	H	H	H	L	L	L	L	H	H	L	X	X
	Electrical Discharge Mach.	L	H	H	H	H	H	L	L	L	H	L	X	X
Sheet Forming Proc.	Sheet Metal	X	H	H	H	H	H	L	L	L	L	X	X	X
	Thermoforming	X	X	X	X	X	X	X	X	X	X	X	H	X
	Metal Spinning	X	H	H	H	H	H	H	L	L	L	L	X	X

The required information is entry in the data structure by the designer, it can thus be reuse by an extraction feature. A factor of compatibility is evaluate according to the previous table (from incompatibility to suitability), which is shown to the user by a set of colors in the data entry feature view (Figure 5.17) or under a graphic as for the decision aided view (Figure 5.15).

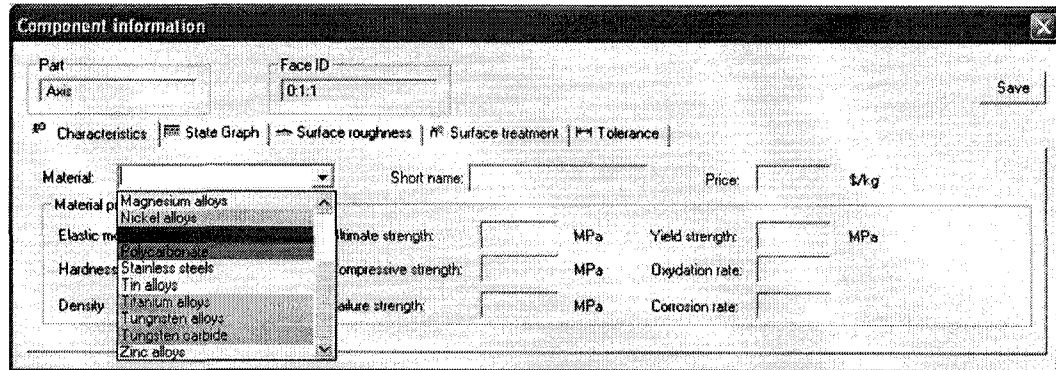


Figure 5.17 User interface using large management of incompatibility

5.6 When add data?

In the previous sections, we have presented a solution to inform the designer about his choices, using technological data integrated to the topological structure. All data are added by various stakeholders or deduce from other parameters using a set of features. The method we proposed dissents from the conventional design method and it increases the workload of users. The risk for designers is to waste time or to loose interest because of overload, instead of help him. The data integration concept should take into consideration the influence of overload and decision aided, to minimize both parameters. The design method should be in the optimal area (Figure 5.18), where the time-saving is highest.

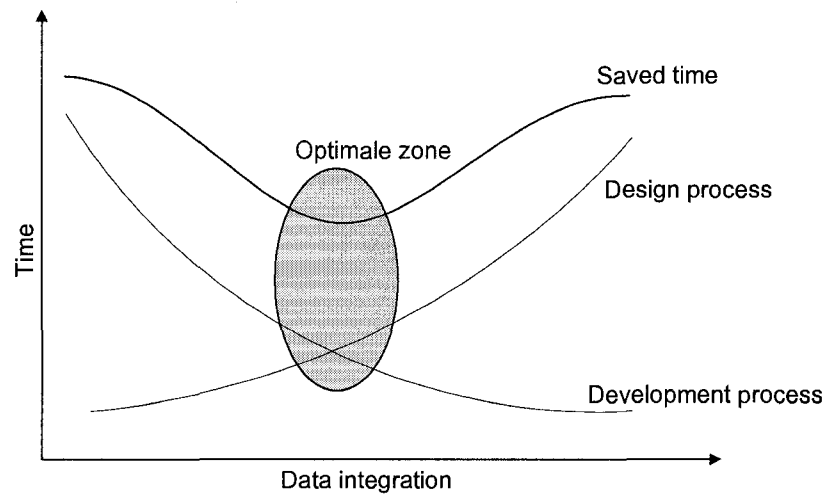


Figure 5.18 Graphical representation of time-saving

5.7 Conclusion

In this paper is shown the importance of integrated design concept to anticipate changes during the development process, increasing production costs. We develop a CAD solution able to store technological information early in the design process and allowing its reuse for decision aided. Such data management enables to save time during the product elaboration, avoiding wasting-time and information redundancy. Our model has been implemented with the C++ object-oriented programming, using OpenCASCADE libraries (display, topological structure development, data base management), to validate the methodology.

Stakeholders are able with this tool to enter technological data and also to adapt their future choices according to there implications on the product lifecycle. Examples developed in this paper can be generalised at every parameter of the product lifecycle. The authors are now studying to support the geometric change avoiding lost of parameters features with the creation of new entities (propagation of information). The data integration should also allowing to automatically generate drafts for manufacturing

process planning using suitable features, in fact needed technological information is ever set in the data structure.

5.8 References

- [1] Boothroyd G, Dewhurst P, Knight WA. Product design for manufacture and assembly. 2nd ed. New York: Marcel Dekker; 2002.
- [2] Miyakawa S, Ohashi T. The Hitachi assemblability evaluation method. Proceedings of the International Conference on Product Design for Assembly, Newport, RI; April 1986.
- [3] El Wakil SD. Processes and design for manufacturing. 2nd ed. Waveland Pr Inc; 2002.
- [4] Tollenaere M. Conception de produits mécaniques : méthodes, modèles et outils. Paris: Hermes; 1998.
- [5] Rehg JA, Kraebber HW. Computer-integrated manufacturing. 2nd ed. Upper Saddle River, N.J.: Prentice Hall; 2001.
- [6] Teissandier D, Couetard Y, Gerard A. Computer aided tolerancing model: Proportioned assembly clearance volume. Comput Aided Des 1999;31(13):805-17.
- [7] Kandikjan T, Shah JJ, Davidson JK. A mechanism for validating dimensioning and tolerancing schemes in CAD systems. Comput Aided Des 2001;33(10):721-37.

- [8] Anselmetti B. Generation of functional tolerancing based on positioning features. *Comput Aided Des* 2006;38(8):902-19.
- [9] Jabbour T, Mascle C, Maranzana R. A database for the representation of assembly features in mechanical products. *Int J Comput Geom Appl (Singapore)* 1998;8(5-6):483-507.
- [10] Lee K, Gossard DC. Hierarchical data structure for representing assemblies: part 1. *Comput Aided Des* 1985;17(1):15-9.
- [11] Noort A, Hoek GFM, Bronsvort WF. Integrating part and assembly modelling. *Comput Aided Des* 2002;34(12):899-912.
- [12] Brissaud D, Paris H, Tichkiewitch S. Assisting designers in the forecasting of surfaces used for easier fixturing in a machining process. *J Mater Process Technol* 1997;65(1-3):26-33.
- [13] Ding X, Zhang L, Lu Y, Jiang R. A Feature Recognition Approach for Mold Machining. 19th International Conference on Computer Applications in Industry and Engineering (CAINE-2006), Las Vegas, Nevada; November 13 - 15, 2006.
- [14] Shah JJ, Rogers MT. Expert form feature modelling shell. *Comput Aided Des* 1988;20(9):515-24.
- [15] Cugini U, Mandorli F, Vicini L. Using features as knowledge formalization for simultaneous engineering. *Proceedings of the IFIP TC5/WG 5.3 Eight International PROLAMAT Conference, Man in CIM - Human Aspects in Computer Integrated Manufacturing, Tokyo, Jpn; Jun 24-26 1992.*

- [16] Chep A, Tricarico L. Object-oriented analysis and design of a manufacturing feature representation. *Int. J. Prod. Res. (UK)* 1999;37(10):2349-76.
- [17] Sy M, Mascle C. Life cycle design and manufacturing approaches. 6th international conference in Integrated Design and Manufacturing in Mechanical Engineering (IDMME06), Grenoble, France; May 17-19, 2006.
- [18] Chang T-C, Wang H-P, Wysk RA. *Computer-aided manufacturing*. 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall; 2006.
- [19] Mandorli F, Rizzi C, Susca L, Cugini U. An approach to implement feature-based applications using knowledge aided engineering technology. *Proceedings of the IFIP TC5/WG5.2 & WG5.3 Conference on Feature Modelling and Advanced Design-for-the-Life-Cycle Systems: Feature Based Product Life-Cycle Modelling*, Valenciennes, France; 12-14 June 2001.
- [20] Shetty D. *Design for product success*. Dearborn, MI: Society of Manufacturing Engineers; 2002.
- [21] Li R-K, Hwang C-L. Framework for automatic DFA system development. *Comput Ind Eng* 1992;22(4):403-13.
- [22] Sturges RH, Jr., Kilani MI. Towards an integrated design for an assembly evaluation and reasoning system. *Comput Aided Des* 1992;24(2):67-79.
- [23] Coma O, Mascle C, Veron P. Geometric and form feature recognition tools applied to a design for assembly methodology. *Comput Aided Des* 2003;35(13):1193-210.

- [24] Coma O, Mascle C, Balazinski M. Application of a fuzzy decision support system in a Design for Assembly methodology. *Int J Computer Integr Manuf* 2004;17(1):83-94.
- [25] Deneux D. Life cycle modelling of an innovation. Proceedings of the IFIP TC5/WG5.2 & WG5.3 Conference on Feature Modelling and Advanced Design-for-the-Life-Cycle Systems: Feature Based Product Life-Cycle Modelling, Valenciennes, France; 12-14 june 2001.
- [26] Haasis S, Layer A, Schulze H. Feature technology: added value for the engineering process - Cost calculation and experience transfer. Proceedings of the IFIP TC5/WG5.2 & WG5.3 Conference on Feature Modelling and Advanced Design-for-the-Life-Cycle Systems: Feature Based Product Life-Cycle Modelling, Valenciennes, France; 12-14 june 2001.
- [27] Bronsvoort WF, Noort A. Multiple-view feature modelling for integral product development. *Comput Aided Des* 2004;36(10):929-46.
- [28] Lee K. Principles of CAD/CAM/CAE systems. Reading, Mass. ; Don Mills, Ont.: Addison-Wesley; 1999.
- [29] Billatos SB, Basaly NA. Green technology and design for the environment. Washington, DC: Taylor & Francis; 1997.
- [30] Koshal D. Manufacturing engineer's reference book. Oxford, UK: Butterworth Heinemann; 1993.
- [31] Ashby MF. Materials selection in mechanical design. 3rd ed. Amsterdam, [The Netherlands]: Elsevier Butterworth-Heinemann; 2005.

CHAPITRE 6 : DISCUSSION GÉNÉRALE

Ce chapitre présente une discussion sur certains aspects méthodologiques et résultats en lien avec la revue de littérature qui n'ont pas été abordés dans les articles. La première section aborde un concept similaire à celui étudié dans ce mémoire pour véhiculer l'information « métier ». Les sections suivantes abordent l'intégration de notre solution dans un milieu industriel en tant qu'échange de données ou d'implémentation logicielle. Les deux dernières sections étudient le transfert de l'information dans le cadre d'une modification géométrique du produit et la contrainte occasionnée par l'intégration des données dans le processus de développement.

6.1 Information « métier » liée au feature durant le cycle de développement

Les *features* sont de plus en plus utilisés dans le domaine de la conception, notamment avec l'introduction du concept FBM. Mais leur utilisation dans le monde industriel reste principalement limitée à l'information géométrique. Cette étude a montré l'importance de l'intégration de l'information technologique grâce aux *features*. Mais d'autres ouvertures sont envisageables dans le domaine de la conception.

En utilisant le concept de vue « métier », où chaque intervenant voit l'information du *feature* selon son domaine de compétence, on peut également faciliter l'étape de la planification de la fabrication (processus de génération de gammes). Par exemple, l'utilisation d'un *feature* « trou » (proposé par la majorité des logiciels actuels) permet de définir une information « métier » dès la conception du produit. Au bureau des méthodes, ce *feature* « trou » comparativement à l'enlèvement de matière traditionnel

(également appelé poche, et qui peut avoir le même aspect géométrique) sera immédiatement associé à une opération de fabrication spécifique, définie par ses outils et toutes les caractéristiques nécessaires à la construction de l'avant projet d'étude de fabrication. L'information contenue par le *feature* « trou » va permettre de déterminer si un lamage ou un chanfrein est requis, si un pointage va être nécessaire, elle va également permettre de déterminer le nombre de passes et la nécessité de l'alésage, etc.

6.2 Intégration dans un contexte industriel

L'utilité de l'application que nous avons développée durant cette étude ne prend sa place dans un contexte de développement collaboratif que dans la mesure où toutes les applications CIM peuvent fonctionner avec ce type de structure de données. Pour faciliter la communication entre tous les logiciels et pour profiter pleinement des avantages de l'intégration de l'information dans la structure de données, cette intégration doit concerner la totalité de l'information relative au produit.

Depuis les débuts de l'informatisation du procédé d'élaboration du produit avec l'introduction de la conception assistée par ordinateur, on a cherché à simplifier la tâche jusque alors manuelle (dessin, calcul, test, maquette, etc) de l'intervenant. Par la suite, c'est tout le reste du processus que l'on a cherché à informatiser, de la conception à la fabrication en passant par la gestion des données. Mais à chaque fois que l'on a créé une solution pour aider un acteur du cycle de vie, on a cherché à implémenter l'ancien modèle sans chercher à réformer la façon de concevoir.

Par exemple, avant l'introduction des systèmes CFAO, le département des méthodes était chargé du tolérancement après la conception. Le modèle de conception de produit actuel dans l'industrie est encore similaire à la méthode pré-informatisée : la mise en plan, gérée automatiquement par toutes les solutions de CAO contemporaines, est toujours présente sous sa forme la plus basique quand elle est transmise sous forme de livrable au département des méthodes. Toutes les informations qui sont ensuite ajoutées

à la mise en plan, le sont uniquement à titre consultatif car elles ne sont pas intégrées à la structure de données. Les logiciels permettent effectivement de rentrer ces données sur la vue 2D, mais elles ne sont pas accessibles numériquement par les utilisateurs chargés de la génération des gammes de production. Pourtant, l'accessibilité à ces données permettrait d'accélérer la génération des gammes de fabrication et d'assemblage.

C'est un non-sens de parler d'intégration des données pour l'industrialisation et de sortir de ce concept à différentes étapes du processus, en se privant de tous les avantages que l'intégration implique.

Le logiciel que nous avons développé peut s'adapter très facilement à n'importe quelle solution de CFAO existante pour peu que ses sources de programmation soient accessibles. En effet, les modules sont programmés en C++ et ne font que construire une base de données des informations en les associant à des entités du produit (face, pièce, assemblage) par l'intermédiaire de sa structure topologique. L'élément essentiel est d'avoir accès à l'information numérique caractérisant ces entités pour pouvoir leur lier les informations respectives (en programmation, on parle de pointeur) et ainsi définir correctement le *feature* technologique. De nombreux outils de gestion du cycle de vie du produit ("*Product Lifecycle Management*" – PLM) permettent déjà de remplir une partie de cette requête, mais ils sont limités à capitaliser l'information concernant uniquement la pièce ou l'assemblage, ce qui permet de faciliter aux utilisateurs l'accès aux données mais pas d'extraire et de traiter automatiquement ces données.

6.3 Échange de l'information

Toujours dans un contexte d'intégration globale avec les autres outils dans le processus d'élaboration du produit, nous nous sommes interrogés sur le format à adopter pour enregistrer les caractéristiques relatives aux *features* technologiques.

Lors de la création d'un des *features* à l'aide du logiciel développé, ses caractéristiques sont stockées dans la mémoire vive de l'ordinateur. L'enregistrement des données est

effectué dans un fichier sous forme d'instructions relatives à chaque type de *feature*. Ces données ne sont cependant pas accessibles directement à un autre logiciel à moins de disposer de la documentation associée pour pouvoir interpréter ces informations.

En matière d'échange de données produit, le format STEP ("*Standard for Exchange of Product data*", ISO/TC 184/SC4) se veut la norme prédominante, c'est un projet ouvert très avancé qui permet de stocker des informations relatives à de nombreux domaines (conception, analyse, fabrication, maintenance, électronique, etc). Il a été défini comme un standard multi-application et multi-utilisation, au format neutre, capable de couvrir tous les stades du cycle de vie de produit. La Figure 6.1 représente un extrait du langage STEP qui montre son principe de fonctionnement.

```

ISO-10303-21;

#10 = ADVANCED_BREP_SHAPE_REPRESENTATION(' ', (#11,#15),#345);

#11 = AXIS2_PLACEMENT_3D(' ',#12,#13,#14);
#12 = CARTESIAN_POINT(' ', (0.E+000,0.E+000,0.E+000));
#13 = DIRECTION(' ', (0.E+000,0.E+000,1.));
#14 = DIRECTION(' ', (1.,0.E+000,0.E+000));

#61=CARTESIAN_POINT('Vertex', (0.,0.,0.)) ;
#63=CARTESIAN_POINT('Vertex', (0.,0.,20.)) ;

#62=VERTEX_POINT(' ',#61) ;
#64=VERTEX_POINT(' ',#63) ;

#69=LINE('Line',#61,#63) ;

#72=EDGE_CURVE(' ',#62,#64,#69,.T.) ;

#87=ORIENTED_EDGE(' ',*,*,#72,.T.) ;

#85=EDGE_LOOP(' ', (#86,#87,#88,#89)) ;

#90=FACE_OUTER_BOUND(' ',#85,.T.) ;

#51=CLOSED_SHELL('Closed Shell', (#91,#122,#153,#175,#187,#199)) ;
#200=MANIFOLD_SOLID_BREP('Corps principal',#51) ;

```

Figure 6.1 Extrait d'un fichier de données utilisant le format STEP

Le langage STEP est basé sur une architecture similaire à celle du B-Rep (point, arrête, contour, face, enveloppe, solide, pièce et assemblage). Le format est neutre et explicite, chaque instruction est exprimée, puis suivie des paramètres qui la caractérise (exemple : l'arrête #69 est définie par les sommets #61 et #63).

On peut donc facilement enregistrer les *features* technologiques utilisés dans cette étude au format STEP pour les exporter et les rendre accessibles aux autres applications. La

syntaxe utilisée sera semblable à celle utilisée dans l'exemple de la Figure 6.2, appliquée à un *feature* technologique :

```
#00 = FEATURE_TECHNOLOGIQUE('nom',
                             #entité de référence,
                             (#paramètre 1,
                              #paramètre 2,
                              #paramètre 3),
                             (#relation 1,
                              #relation 2),
                             ) ;
```

Figure 6.2 Exemple de *feature* technologique au format STEP

De plus, la plateforme de développement logiciel OpenCASCADE que nous avons utilisée autorise l'exportation au format STEP, ce qui évite à l'utilisateur d'utiliser un logiciel de conversion supplémentaire.

6.4 Discussion sur le suivi de l'information selon l'évolution géométrique du produit

L'ingénierie simultanée permettant l'avancée de plusieurs étapes de la conception en parallèle, la géométrie du produit doit être considérée comme évolutive au cours du processus de développement. Certaines entités du produit ont peut être déjà reçues de l'information technologique pour les étapes de fabrication et d'assemblage, mais leur géométrie n'est pas encore définitive et peut être modifiée. Il s'agit donc de s'interroger sur la manière d'assurer le suivi des données technologiques liées à une entité durant ces changements (Figure 6.3). L'entité principalement concernée dans ce cas est la face.

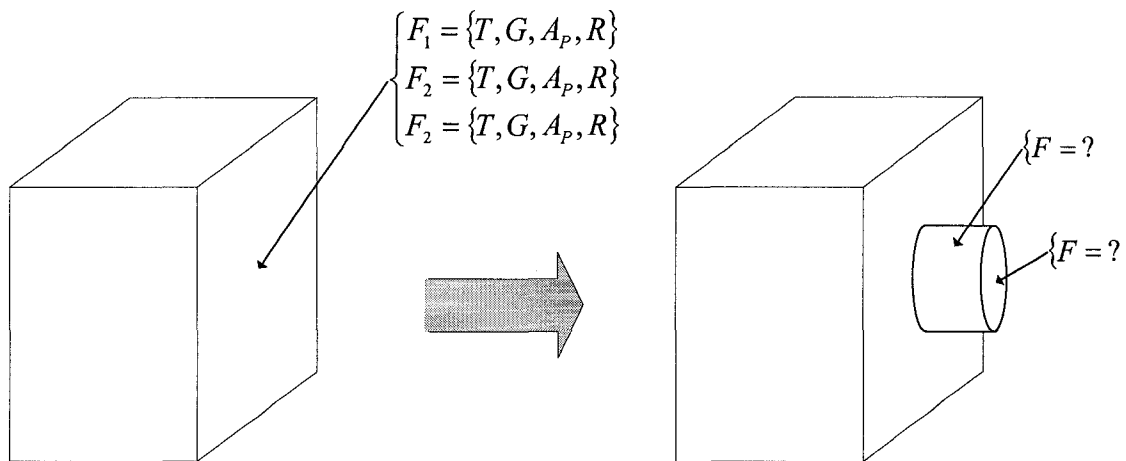


Figure 6.3 Suivi de l'information avec l'évolution géométrique du produit

Plusieurs cas se présentent, et ils sont souvent liés entre eux (l'un peut impliquer l'autre) :

- Entité créée (entité initiale)
- Entité modifiée (changement des propriétés de l'entité)
- Entité générée (le changement découle de la modification d'une autre entité)
- Entité supprimée

Dans le cas où une nouvelle face est créée, et si elle n'a jamais été modifiée auparavant, on est dans le cas d'une entité initiale. Les *features* de cette entité n'existent donc pas sauf si l'utilisateur a décidé, en raison de ses connaissances sur le produit, de prérégler certaines informations (c'est le cas quand l'entreprise conçoit un nouveau produit en se basant sur la conception d'un ancien produit et les enseignements qu'elle en a tiré).

Dans le cas où l'entité est supprimée, elle disparaît de la structure de données et les *features* associés sont également déduits.

Si les propriétés d'une entité ont été modifiées, les propriétés des *features* restent inchangées. L'entité étant toujours la même, le pointeur de référence reste identique

donc le *feature* n'est pas modifié. L'utilisateur peut en revanche modifier les paramètres de *feature* s'il le désire. Si une des propriétés du *feature* dépendait de la géométrie, alors le paramètre correspondant est mis à jour ou supprimé en cas d'incompatibilité. Exemple : pour un *feature* d'usinage relatif au perçage, la longueur de l'alésage ne peut excéder un rapport défini entre la profondeur et le diamètre au-delà duquel l'opération est techniquement irréalisable.

Dans le cas où le changement entraîne la création de plusieurs entités (mis en évidence dans la Figure 6.3), deux cas se présentent :

- Le *feature* concerné est caractérisé par une contrainte de propagation, qui entraîne la création de ce *feature* pour une partie ou la totalité des entités filles avec les mêmes propriétés que l'entité mère.
- Le *feature* concerné n'est pas destiné à se propager, alors les entités créées sont traitées en tant que entité initiale. Elles sont donc exemptes d'information technologique.

Dans tous les cas présentés ci-dessus, les *features* de l'entité mère gardent à chaque fois leurs propriétés initiales (choix par défaut), mais le concepteur peut avoir à supprimer certaines propriétés devenues obsolètes. Exemple : un *feature* de qualité de surface associé à une entité mère, dont la modification entraîne un transfert de la surface fonctionnelle vers l'entité fille, devient inutile. C'est l'entité fille, devenue surface fonctionnelle, qui hérite de cette propriété du *feature*.

Dans certains cas d'entités générées, il peut arriver que l'entité mère disparaisse, on est alors confronté aux cas précédents pour les entités filles, mais avec une disparition de l'entité mère et une suppression des *features* qui lui sont associés.

6.5 Gêne de l'utilisateur

Dans le chapitre 4, nous avons abordé la notion de gêne de l'utilisateur du logiciel développé. En effet, la notion d'intégration de données présentée dans cette étude tend à alléger la charge des ingénieurs méthode en ajoutant les informations technologiques dès l'étape de conception. Le concepteur est donc en charge d'une plus grande quantité de données. Le risque de cette méthode de conception est de voir le concepteur négliger l'une de ses tâches devant l'ampleur et/ou la répétitivité du travail.

Pour être efficace et permettre l'automatisation de certaines tâches et l'extraction d'informations, l'intégration de données doit donc être complétée au maximum. Il convient cependant de s'interroger sur le moment où les données sont utiles au cours du développement, afin de déterminer une décharge du concepteur : certaines informations peuvent être entrées par le département méthode sans que cela ait de conséquences sur l'efficacité du procédé de développement. C'est le cas d'une partie des informations concernant la fabrication qui ne sont pas utiles avant la génération des gammes de fabrication.

L'efficacité du concepteur est également influée par l'ergonomie de l'interface utilisateur du logiciel. La quantité importante de paramètres définissant un produit et ses processus de fabrication peut rapidement devenir sources de confusion. Nous avons donc cherché à simplifier cette interface au maximum en donnant à l'utilisateur une information codifiée. C'est ainsi que les indices de compatibilité et d'efficience sont présentés au chapitre 5 sous forme de couleurs, permettant aisément leur identification. La précision des facteurs d'efficience n'était pas prépondérante dans cette étude car l'objectif principal était de fournir un moyen de comparaison explicite et rapide des différents choix de conception.

6.6 Conclusion

Dans ce chapitre, nous avons discuté de la possibilité d'implémenter notre logiciel aux solutions CAO actuelles ainsi que des capacités du modèle à communiquer avec les autres applications CIM. Nous avons également étudié l'impact sur l'utilisateur et la propagation des informations confrontées à l'évolution géométrique inhérente au concept d'ingénierie collaborative.

La conclusion de ce mémoire présente les contributions de l'étude à l'avancement des connaissances, en mettant en valeur les limites et contraintes rencontrées. Elle expose également les recommandations et perspectives de la recherche dans ce domaine.

CONCLUSION

La maîtrise de l'information sur le produit et la gestion de la communication entre les acteurs du cycle de développement de produit sont les facteurs prépondérants dans le contexte d'ingénierie simultanée. Le besoin grandissant de prendre en compte dès la conception les impacts liés aux procédés de production, mais également liés à des concepts plus exotiques comme le développement durable ou les facteurs humains, ont rendu l'intégration des données nécessaire au processus d'élaboration d'un produit. On utilise déjà des solutions logicielles pour gérer les données techniques, les relations clients, les ressources de l'entreprise, mais pas pour gérer les données relatives à la face.

La structure de données développée dans l'article "Data structure applied to product lifecycle" a permis d'améliorer l'intégration des données techniques au sein de la maquette numérique :

- gain de temps lors de l'entrée des données, grâce à une vue dédiée qui centralise toutes les opérations,
- accessibilité de l'information tout au long du cycle de développement du produit,
- données attachées directement à leur entité correspondante dans la base de données.

L'accessibilité de l'information était l'objectif principal de la première partie de l'étude, qui nous a permis de traiter les données relatives au produit. L'article "Assisting designer using feature modeling for lifecycle" s'est donc basé sur cette structure de données afin d'aider le concepteur dans sa tâche. L'application développée dans ce cadre permet de :

- caractériser la viabilité des choix du concepteur en temps réel afin de lui donner un retour d'information des impacts de ses solutions sur le cycle de développement.
- informer le concepteur sur le degré d'incompatibilités des choix qu'il envisage.

Les perspectives de ce travail sont très nombreuses. En effet, l'intégration totale des données est un concept qui permet d'assister ou d'automatiser la plus grande partie des tâches manuelles des intervenants. Les principales applications découlant de cette accessibilité des données concernent l'aide à la conception.

La première perspective concerne directement les résultats du deuxième article, consacré à l'optimisation du processus de développement. Comme évoqué en introduction, l'ouverture de ce concept au *Design For X* permettrait de contrôler les paramètres de l'ensemble du cycle de vie dès l'étape de la conception

Il serait très aisé de mettre en place un filtre sur les données de *features* pour présenter une interface complètement orientée « métier », ce qui permettrait de simplifier le travail des utilisateurs : les informations visibles à l'écran ne concerneraient que leur domaine d'application.

On peut également imaginer une application chargée de prendre en charge le savoir faire de l'entreprise. Un *feature* pourrait être chargé de prérégler certains paramètres des *features* appliqués aux entités du produit pour accélérer la conception, en se basant sur les produits déjà conçus.

L'extraction des données peut également servir à améliorer la création des gammes de fabrication et d'assemblage. La séquence des états de fabrication est directement liée aux relations entre chaque face déterminée par la cotation fonctionnelle. L'ordre de cotation des faces, défini par le concepteur, peut être enregistré par un *feature* afin de transmettre l'information au département des méthodes.

Enfin, nous pouvons encourager une réflexion sur la mise en plan en 2 dimensions. Cette étape est encore inévitable dans le processus de conception (ne serait ce que pour la certification des produits) même si l'on cherche actuellement à se passer de ce support. En effet, l'intégration des données étant complète (toutes les entités topologiques du produit sont couvertes) la vue 2D n'apporte plus aucune information : la cotation peut aisément être effectuée directement sur la maquette numérique, ainsi que toutes les opérations de définition des surfaces. On dispose en tout cas de toutes les informations nécessaires dans la structure topologique pour générer cette vue automatiquement.

BIBLIOGRAPHIE

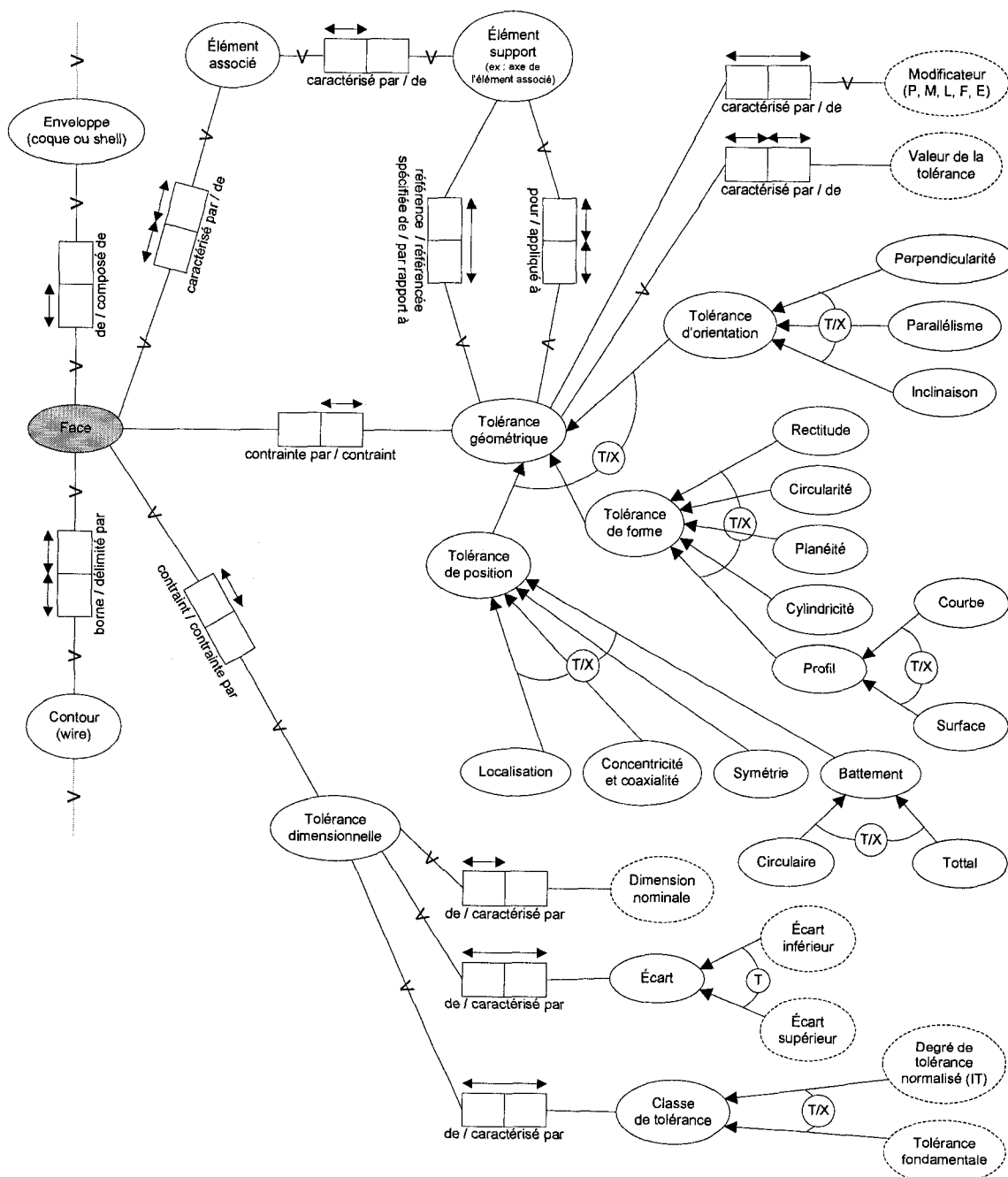
- Anwer, N., Mathieu, L., & Elmaraghy, W. (2005). *Apport des ontologies de spécification et de vérification géométrique des produits pour la conception intégrée*. Congrès de conception et production intégrées, Casablanca, Maroc.
- Billatos, S. B., & Basaly, N. A. (1997). *Green technology and design for the environment*. Washington, DC: Taylor & Francis.
- Boothroyd, G., Dewhurst, P., & Knight, W. A. (2002). *Product design for manufacture and assembly* (2nd ed.). New York: Marcel Dekker.
- Brissaud, D. (1992). *Système de conception automatique de gammes d'usinage pour les industries manufacturières*. Thèse de doctorat, Université Joseph Fourier - Grenoble 1.
- Brissaud, D., Paris, H., & Tichkiewitch, S. (1997). Assisting designers in the forecasting of surfaces used for easier fixturing in a machining process. *Journal of Materials Processing Technology*, 65(1-3), 26-33.
- Chang, T.-C., Wang, H.-P., & Wysk, R. A. (2006). *Computer-aided manufacturing* (3rd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.
- Chep, A., & Tricarico, L. (1999). Object-oriented analysis and design of a manufacturing feature representation. *International Journal of Production Research*, 37(10), 2349-2376.
- Cugini, U., Mandorli, F., & Vicini, L. (1992, Jun 24-26 1992). *Using features as knowledge formalization for simultaneous engineering*. Proceedings of the IFIP TC5/WG 5.3 Eight International PROLAMAT Conference, Man in CIM - Human Aspects in Computer Integrated Manufacturing, Tokyo, Jpn.

- Cunningham, J. J., & Dixon, J. R. (1988, Jul 31-Aug 4 1988). *Designing with features: The origin of features*. Computers in Engineering, San Francisco, CA.
- De Jong, F., Reitsema, T., & Hoogeboom, R. (1992). Feature based modelling: a way to improve the CAD/CAM-process. *Revue internationale de C.F.A.O. et d'infographie*, 7(2), 209-234.
- Deneux, D. (2002). *Méthodes et modèles pour la conception concourante*. Thèse HDR, Université de Valenciennes et du Hainaut Cambrésis.
- Dixon, J. R., Cunningham, J. J., & Simmons, M. K. (1987, 6-8 Oct. 1987). *Research in designing with features*. Intelligent CAD, I. Proceedings of IFIP TC/WG 5.2 Workshop on Intelligent CAD, Boston, MA.
- Dong, X., DeVries, W. R., & Wozny, M. J. (1991, Aug 18-24 1991). *Feature-based reasoning in fixture design*. 41st General Assembly of CIRP, Palo Alto, CA.
- Dong, X., & Wozny, M. (1990, Aug 5-9 1990). *Managing feature type dependency in a feature-based modeling system*. Proceedings of the 1990 ASME International Computers in Engineering Conference and Exposition, Boston, MA.
- Duverlie, P. (1996). *Study and proposition of a technical cost estimation method applied to mechanical manufacturing and based on case based reasoning*. Thèse de doctorat, Université de Valenciennes, Valenciennes.
- Gardan, Y. (2003). CAO : vers la modélisation fonctionnelle. *Techniques de l'Ingénieur* (Vol. H 3 752, pp. 20).
- Grayer, A. R. (1976). *A computer Link Between Design and Manufacturing*. Thèse de doctorat, University of Cambridge.
- Habrias, H. (1990). La méthode Niam et son symbolisme. *Bulletin de l'EPI*, 51, 219-229.

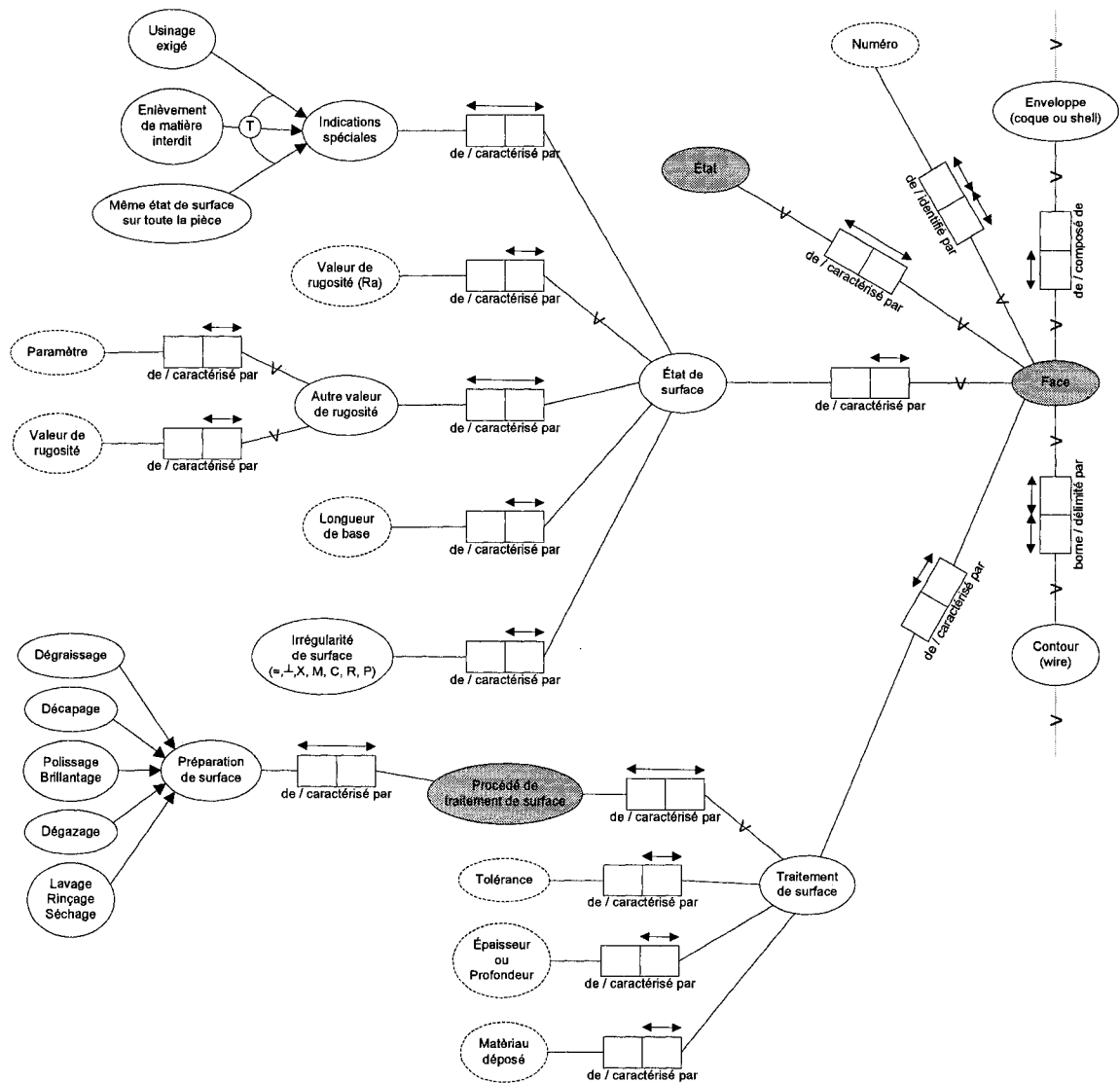
- Halpin, T. (1995). *Conceptual schema and relational database design* (2nd ed.). Sydney: Prentice Hall Australia.
- Jabbour, T. (1997). *Intégration de modèles de caractéristiques à une modélisation géométrique pour supporter l'assemblage*. Thèse de doctorat, Ecole Polytechnique de Montréal.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2007). Disassembly scheduling: Literature review and future research directions. *International Journal of Production Research*, 45(18-19), 4465-4484.
- Lee, K. (1999). *Principles of CAD/CAM/CAE systems*. Reading, Mass. ; Don Mills, Ont.: Addison-Wesley.
- Martin, P. (1999, juillet 1999). *Ingénierie simultanée : conception des processus de fabrication*. Université d'été GAMA, E.N.S. Cachan.
- Masclé, C. (1990). *Approche méthodologique de détermination de gammes par le désassemblage*. Thèse de doctorat, Ecole polytechnique fédérale de Lausanne.
- Msaaf, O. (2002). *Validation des caractéristiques d'usinage par des grammaires d'usinage attribuées: Une contribution à la prise en compte des contraintes d'usinage en cours de conception*. Ph.D., Ecole de Technologie Supérieure (Canada).
- OpenCASCADE. (2007). Open CASCADE Technology: OpenCASCADE S.A.S.
- Paoluzzi, A., & Pascucci, V. (2003). *Geometric programming for computer-aided design*. Hoboken: John Wiley & sons.
- Shah, J. J. (1991). Assessment of features technology. *Computer Aided Design*, 23(5), 331-343.

- Shah, J. J., & Rogers, M. T. (1988). Expert form feature modelling shell. *Computer Aided Design*, 20(9), 515-524.
- Shetty, D. (2002). *Design for product success*. Dearborn, MI: Society of Manufacturing Engineers.
- Sky, R. (1996, December 1996). *Team Collaboration: A report on the Spacecraft System Design course offered through the University of Western Ontario Undergraduate Engineering program*. CEAMRL Report 96_03, University of Western Ontario, Canada.
- Sy, M., & Mascle, C. (2006, May 17-19, 2006). *Life cycle design and manufacturing approaches*. 6th international conference in Integrated Design and Manufacturing in Mechanical Engineering (IDMME06), Grenoble, France.
- Tollenaere, M. (1998). *Conception de produits mécaniques : méthodes, modèles et outils*. Paris: Hermes.
- Van Houten, F. J. A. M., & Van 't Erve, A. H. (1992, 10-12 Aug. 1992). *Feature based manufacturing with PART*. Proceedings of the International Conference on Manufacturing Automation, Hong Kong.

ANNEXE A : TOLÉRANCEMENT DE L'ENTITÉ FACE



ANNEXE B : DÉFINITION TECHNOLOGIQUE DE L'ENTITÉ FACE



ANNEXE C : CARACTÉRISTIQUES DU FEATURE DE TRAITEMENT DE SURFACE

